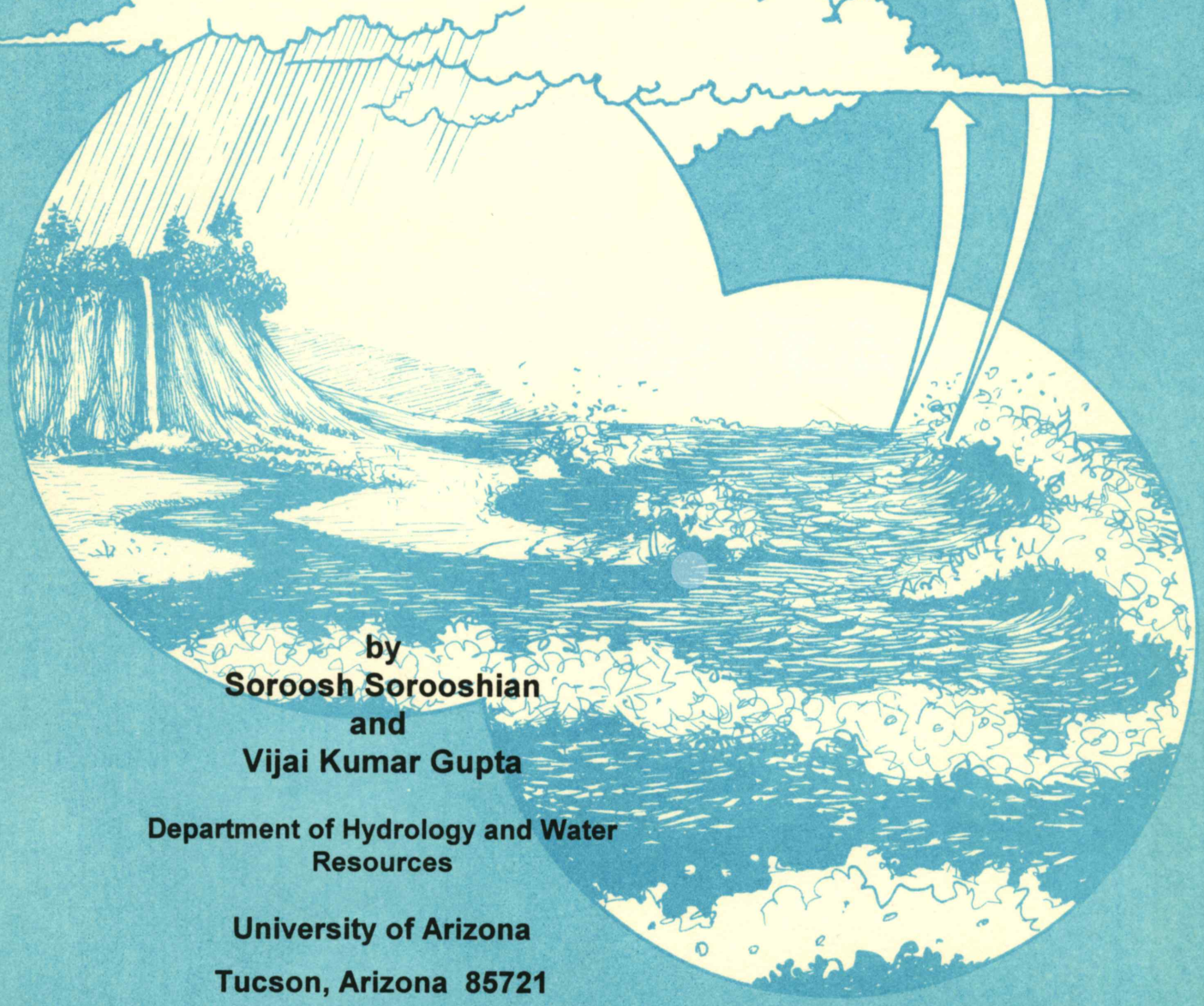
HWR No. 86-010

Improving the Reliability of Compartmental Models: Case of Conceptual Hydrologic Rainfall-Runoff Models



by
Soroosh Sorooshian
and
Vijai Kumar Gupta

Department of Hydrology and Water Resources

University of Arizona

Tucson, Arizona 85721

HWR No. 86-010

Improving the Reliability of Compartmental Models: Case of Conceptual Hydrologic Rainfall-Runoff Models

by
Soroosh Sorooshian
and
Vijai Kumar Gupta

Department of Hydrology and Water Resources

University of Arizona

Tucson, Arizona 85721

Research Project Report

on

IMPROVING THE RELIABILITY OF COMPARTMENTAL MODELS: CASE OF CONCEPTUAL HYDROLOGIC RAINFALL-RUNOFF MODELS

Submitted to

National Science Foundation

Grant No. CEE-8217823

by

Soroosh Sorooshian
Principal Investigator
Associate Professor
Department of Hydrology and Water Resources

and

Vijai Kumar Gupta
Post-Doctoral Research Associate

University of Arizona
Tucson, Arizona 85721

August 1986

ACKNOWLEDGMENTS

The Principal Investigator wishes to acknowledge the technical support provided by the Center for Computing Information and Technology at the University of Arizona. The typing and editing of this report was done by Ms. Corla Thies. Her patience and creativity are greatly appreciated.

LIST OF PUBLICATIONS AS THE RESULT OF THIS RESEARCH PROJECT

Gupta, V.K., and S. Sorooshian, "The Automatic Calibration of Conceptual Catchment Models Using Derivative-Based Optimization Algorithms," Water Resources Research, 21(4), pp. 473-486, April 1985.

Sorooshian, S., and V.K. Gupta, "The Analysis of Structural Identifiability Theory and Application to Conceptual Rainfall-Runoff Models," Water Resources Research, 21(4), pp. 487-496, April 1985.

Gupta, V.K., and S. Sorooshian, "The Relationship Between Data and the Precision of Estimated Parameters," Journal of Hydrology, 81, pp. 57-77, 1985.

Sorooshian, S., "Synthesis of Hydrologic and System Sciences in the Development of Rainfall-Runoff Models," Invited paper for special issue on system identification and parameters. In Revue des Sciences de l'eau, 1(1/4), pp. 21-28, 1985.

Sorooshian, S., "Synthesis of Hydrologic and System Sciences in the Development of Rainfall-Runoff Models," Invited paper for special issue on system identification and parameters. In Hydrology and Environment for Applied Mathematics and Computation: Modeling the Environment, 17, pp. 279-298, 1985.

Sorooshian, S., "The Impact of Catchment Modeling on Hydrologic Reliability," NATO Advanced Institute in "Engineering Reliability and Risk in Water Resources," Tucson, Arizona, May 19-June 2, 1985, Edited by L. Duckstein and E. Plate, to appear.

Gupta, V.K., and S. Sorooshian, "The Influence of Data Length, Information Content, and Noise Characteristics on Model Calibration," In: Multivariate Analysis of Hydrologic Processes, Edited by H.W. Shen et al., Published by Colorado State University Press, Fort Collins, Colorado, pp. 434-449, 1986.

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

TABLE OF CONTENTS--Continued

TABLE OF CONTENTS--Continued

LIST OF TABLES

LIST OF TABLES--Continued

LIST OF FIGURES

LIST OF FIGURES--Continued

CHAPTER 1

CONCEPTUAL MODELING OF THE RAINFALL-RUNOFF PROCESS

1.1 Introduction

The task of developing models that can accurately predict the surface runoff of water from a watershed area subjected to precipitation is one of the major concerns of the hydrologic research community. Such models are useful both for the development of long-term strategies to manage surface water resources and for the short-term prediction of flood events. This topic has therefore been the focus of extensive research. It has led to the development of a whole host of mathematical models that attempt to represent either a part or the whole of the watershed rainfall-runoff process.

Due to its extreme complexity, the rainfall-runoff process has not proved to be easy to model. As emphasized by Dracup et al. (1973), the rainfall-runoff process is a highly nonlinear, time variant, spatially distributed process, and there is probably no available model that adequately captures these characteristics. Among the techniques available which may be used to model a dynamical physical process is the so-called "conceptual" approach. In this approach, the internal operation of the process is described in terms of a number of interlinked subsystems, each representing a different aspect of the process. Since each subsystem module can be independently developed by drawing upon the best knowledge of the physical process that is available, very complex and realistic models of the physical process can be developed. This report deals with the conceptual approach to the identification of a model of the rainfall-runoff process.

1.2 The General Identification Problem

1.2.1 Introduction

Consider a physical process $\mathscr{P}$ (e.g., the watershed rainfall-runoff process) for which certain characteristics such as inputs 'u' (e.g., precipitation) and outputs 'z' (e.g., streamflow, evapotranspiration) can be observed. The General Identification Problem can be broadly stated as that of finding a model $\mathscr{M}$ of the process $\mathscr{P}$ that operates on the inputs 'u' to reproduce the outputs 'z', and that can be manipulated or examined with a view to making statements about the actual process $\mathscr{P}$. In order to solve the General Identification Problem, the researcher must choose from among all the various types of models which might potentially be used to model the process. One of the most useful classes of models (in this age of the digital computer) is the mathematical model. We shall restrict our attention to this class and, hereafter, the word "model" should be taken to mean "mathematical model" unless otherwise stated. Let us define the following sets and terminology:

$\mathscr{U}$ = the universe of all possible models (mathematical and non-mathematical) (note that $\mathscr{P} \in \mathscr{U}$);

$\mathscr{M}$ = the set of all (mathematical) models ($\mathscr{M} \subset \mathscr{U}$);

$\mathscr{M}_\theta$ = the set of all parametric models ($\mathscr{M}_\theta \subset \mathscr{M}$);

$M(\underline{\theta})$ = the set of all elements of $\mathscr{M}_\theta$, restricted to a particular structure, where the parameters θ are restricted to vary within a specified set Θ. Without loss of generality, the parameters will be assumed to be time invariant. For practical purposes, the number of elements in $\underline{\theta} = \theta_1, \ldots \theta_p^T$ will be finite. ($M(\underline{\theta}) \subset \mathscr{M}_\theta$); and

$M(\tilde{\theta})$ = a particular model obtained from the set $M(\underline{\theta})$ by restricting $\underline{\theta} = \underline{\tilde{\theta}} \in \Theta$. $(M(\tilde{\theta}) \in M(\theta))$.

1.2.2 The Practical Identification Problem

There are two major stages in the solution of the General Identification Problem. The first stage consists of the identification of a suitable "structure" (i.e., the mathematical form of the relationships among the inputs, outputs, and other process variables) for the model so as to restrict the class of models to the set $M(\underline{\theta})$. The second stage then consists of selecting a particular model $M(\underline{\tilde{\theta}}) \in M(\underline{\theta})$ which is in some sense "closest" to $\mathscr{P}$.

It is important to recognize that the degree of success that may be achieved in the solution of the identification problem depends in great measure on a careful treatment of Stage 1. Unfortunately, the treatment of Stage 1 is generally more difficult than Stage 2 since the former is not as amenable to mathematical treatment as is the latter. In a few special cases, methodologies have been developed to help select among different model structures (see e.g., Box and Jenkins, 1976; Akaike, 1974; Kashyap, 1982), but these are in general restricted to the treatment of special classes of linear systems.

The methodology used to deal with Stages 1 and 2 depends significantly on the nature of the "information" available about the process $\mathscr{P}$. This information can take two forms:

(a) Qualitative Information

(b) Quantitative Information

The amount of information of each type available helps determine the procedure used for the model identification. In general, the qualitative information is used in Stage 1 to help in restriction of the model set, while the quantitative information is used in Stage 2 to help select one among the many models contained in the set. It is important to note that the interpretation of the qualitative information involved in Stage 1 is to a large extent a subjective process, since the information itself is descriptive and incomplete and therefore "fuzzy" (e.g., Bellman and Zadeh, 1970), and hence open to numerous interpretations. In the case of the watershed process, this information is in the form of descriptions (and sometimes physical laws) of the various processes that occur in a watershed and that influence the mechanisms by which the precipitation over a watershed is converted into groundwater storage and streamflow. Some of the processes, such as the subsurface flow of groundwater, have well-established physical laws relating water movement to the physical properties of the soil. Others, however, such as rainfall interception and potential evapotranspiration, are not as well defined.

Although we would like to build as complex and detailed a model as our available qualitative information would permit, from a practical standpoint this is rarely, if ever, feasible. The major reason for this is that, unfortunately, the level of sophistication of the mathematical tools available for treatment of Stage 2 is insufficient to effectively deal with each and every type of model that might be developed in Stage 1. The requirements of mathematical tractability imposed by the procedures available for use in Stage 2 usually make it necessary to simplify the "conceptualization" of the process. In the case of watershed modeling, this usually takes the form of spatial lumping of the watershed process and the use of simple

semi-empirical equations to represent the complex nonlinear process interactions. This has given rise to the class of watershed models sometimes referred to as "Conceptual Rainfall-Runoff Models" in the literature (for examples, see Section 1.3). It is important to bear in mind that, although these models are referred to as conceptual, they have been developed by trading off some degree of conceptual realism to gain mathematical tractability.

As the preceding discussion indicates, there is no clearly defined methodology for the selection of an appropriate model set. The treatment of Stage 1 remains, therefore, more of an art than a science. In certain applications, such as industrial process control, the model set is restricted to that of linear models, largely due to considerations regarding its relative mathematical tractability. Such approaches have also been extensively studied for use in watershed modeling (for a discussion on the state-of-the-art, see Kibler and Hipel, 1979; Sorooshian, 1983), and their application has met with varied degrees of success. In general, however, it is desirable that the model set be capable of reproducing much of the dominant behavior observed to be characteristic of the process $\mathscr{P}$, and in any case should not have characteristics which run counter to the facts contained in our store of qualitative information about the process.

Once a particular model set has been chosen through Stage 1, we are then faced with the problem of selecting a specific model within this set to represent the particular process of interest (e.g., the watershed). This problem of model selection is equivalent to the selection of the parameter set $\underline{\theta} \in \Theta$ for which the model $M(\underline{\theta})$ is in some sense "closest to the process $\mathscr{P}$."

A geometric interpretation of this concept is presented in Figure 1.1, using the set notation introduced in Subsection 1.2.1. Note that the process $\mathscr{P}$ may be contained in any of the subsets $M(\underline{\theta})$, $\mathscr{M}_{\theta}$, and $\mathscr{M}$ of $\mathscr{U}$ or it may not. If the process is a member of $M(\underline{\theta})$ as is for example the process $\mathscr{P}_1$, then it is conceptually possible to find the "closest" model M_1 to be equivalent to the process $\mathscr{P}_1$ itself. In this case, the model set $m(\underline{\theta})$ is said to contain the process $\mathscr{P}_1$, and the identification problem is equivalent to finding the "true" parameters "$\underline{\theta}_{true}$" of the process. An example of this might be a simulation study to test the efficiency of a parameter estimation methodology.

Similarly, in certain cases, the process $\mathscr{P}$ (see $\mathscr{P}_2$) may be a member of the set $\mathscr{M}_{\theta}$, which is broadly the collection of all linear and nonlinear parametric models. Our interest may lie in identifying the "closest" model M_2 restricted, for example, to the set $M^{\ell}(\theta)$ of all linear models of parameter dimension p. Examples of these are the so-called "Reduced-Order Models," or "Reduced Complexity Models," (e.g., Mahmoud and Singh, 1981). In other cases, the process $\mathscr{P}$ (see $\mathscr{P}_3$) may be contained in the set $\mathscr{M}$ of all possible mathematical models, and the identification problem may be to find $M_3 \in M(\underline{\theta})$ which is closest to $\mathscr{P}_3$ (parametric representation of a nonparametric process).

Finally, the most common situation, and the one of practical interest, occurs when the process $\mathscr{P}$ (see $\mathscr{P}_4$) lies entirely outside of the set $\mathscr{M}$ of mathematical models, and we are interested in finding $M_4 \in M(\underline{\theta})$ which is "closest" to $\mathscr{P}_4$. The problem of modeling the watershed rainfall-runoff process belongs clearly to this category. As the geometric interpretation of Figure 1.1 suggests, the more realistic is the selected

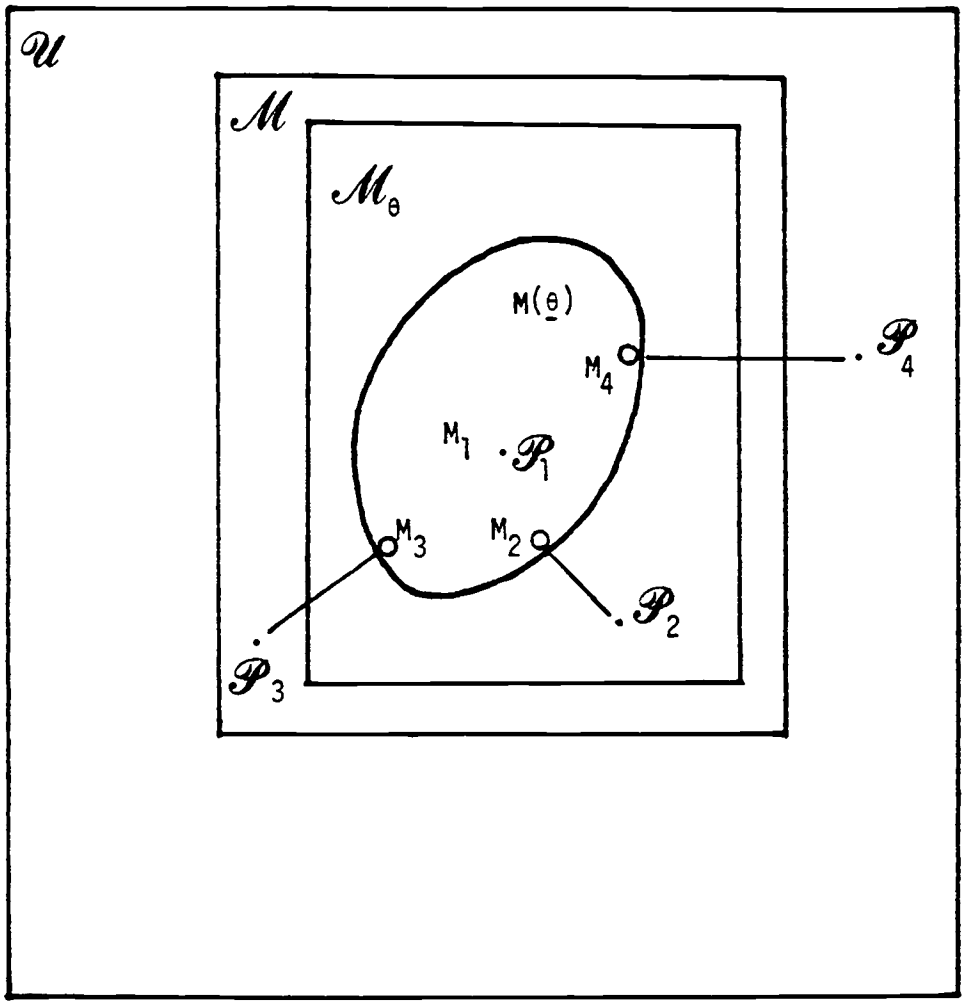


Figure 1.1. Geometric interpretation of the identification problem

model set, the closer will our identified model be to the true process $\mathcal{P}$. Hence, the closest we can potentially come to reproducing the behavior of the process (watershed) is conditional on our ability to select an appropriate model set in Stage 1. The closest we do actually come is conditional on our ability to solve the parameter estimation problem in Stage 2. Note here that the meaning of "closeness" has yet to be suitably defined.

1.2.3 Practical Considerations

In order to successfully solve the identification problem described above, a methodology is required that recognizes the strengths and limitations of the mathematical tools available to us. Clearly, the model set should be carefully chosen to ensure that we have a chance of solving the parameter estimation problem while retaining as much similarity to the process as possible. On the other hand, the parameter estimation methodology chosen should be capable of handling the degree of complexity of the problem while bearing in mind the constraints imposed by availability of computing power. In order to arrive at a successful solution to the identification problem, the following issues must be suitably addressed:

(a) A suitable model set must be chosen whose parameters are identifiable based on the kind of information and mathematical tools available to us;

(b) The measure of closeness must be carefully defined;

(c) A suitably informative data set must be selected;

(d) A methodology for searching for the "closest" model based on (b) and (c) must be developed; and

(e) A methodology for evaluating the degree of success of the identification procedure must be specified.

In the next section, the state-of-the-art in the conceptual modeling of watershed rainfall-runoff processes is reviewed.

1.3 Review of the Literature

1.3.1 The Development of Conceptual Models

The earliest known attempt to develop a conceptual model of the watershed process was reported by Linsley and Ackermann (1942). Since then, numerous other models have been reported in the literature. Rockwood (1958) developed the Streamflow Synthesis and Reservoir Regulation (SSARR) model that has been widely used by the Corps of Engineers. Dawdy and O'Donnell (1965) developed a nine-parameter model with two moisture storages which contained many of the features such as threshold elements found in models more popular today. Boughton (1965, 1966) reported on the development of a model for predicting the response of watersheds such as those found in Australia. Crawford and Linsley (1966) developed the Stanford Watershed Model, which is one of the best known today, and which contains one of the most detailed descriptions of the watershed process that are available. Various modifications to this model have been reported by Liou (1970), Claborn and Moore (1970), Crawford (1971), and Ricca (1972). Probably the most sophisticated outgrowth of the Stanford Watershed Model is the Hydrocomp Simulation Program (Crawford, 1971) which includes water quality simulation capabilities. Holtan and Lopez (1973) described the development and revision of the USDAHL model which is primarily designed to serve the purposes of agricultural watershed engineers. The U.S. National Weather Service currently

uses a conceptual soil moisture accounting model known by the acronym SMA-NWSRFS, which contains less internal detail than the Stanford Watershed Model and is specifically designed for flood forecasting on larger watersheds (Burnash et al., 1973). Modified versions of the SMA-NWSRFS that are designed for the application of feedback techniques to improve forecast performance have been reported by Kitanidis and Bras (1978) and Goldstein and Larimore (1980). Other models that have received widespread application or attention include the USGS model (Carrigan, 1973) designed for short-event simulation and the HEC-1 model (U.S. Army Corps of Engineers Report, 1973) designed as a computational aid for reservoir management and/or flood hazard assessment. Recently, a new approach to watershed modeling which attempts to explicitly account for the spatial variability in a watershed was proposed by Moore and Clarke (1981). Detailed discussions of many of the models mentioned above can be found in Fleming (1975) and Viessman et al. (1977).

1.3.2 Specification of the Measure of Closeness

The measure of closeness most often used for the identification of conceptual rainfall-runoff watershed model parameters has been the Simple Least Squares (SLS) function (e.g., Dawdy and O'Donnell, 1965; Chapman, 1970; Murray, 1970, Johnston and Pilgrim, 1976; Pickup, 1977, Mein and Brown, 1978; among others). Various other measures have also been proposed and examined in the literature, most being based on subjectively chosen transformations of the data so as to give greater weightage to the fitting of high or low level outputs (e.g., Beard, 1967; Dawdy and Lichty, 1968; Chapman, 1970, Johnson and Pilgrim, 1976; Garrick et al., 1978; Manley, 1978; among others). The subjective nature of these criteria has been pointed out in

the literature. Clarke (1973) stated that the input and output data contain measurement error and that the statistical properties of these errors should be recognized while selecting an appropriate measure of closeness. He noted that, when the output data contain measurement error, the use of the SLS measure is optimal only if the errors are independently distributed random variables with zero mean and constant variance. A number of researchers have reported that the use of the SLS as a measure of closeness resulted in conceptually unrealistic values for the model parameters (e.g., Murray, 1970; Ibbitt, 1970; Monro, 1971; Sorooshian, 1978; Sorooshian, Gupta, and Fulton, 1983). Many researchers attempted to resolve this problem by imposing subjectively chosen constraints on the values that the parameters might achieve.

Sorooshian (1978) proposed the use of Maximum Likelihood theory as a means of explicitly accounting for the stochastic nature of the data in the selection of an appropriate measure of closeness. He assumed that the distribution of the output errors could be approximated by a Gaussian probability density and considered the effects of various assumptions regarding the covariance structure of the errors. Measures of closeness were developed for the cases of (a) first-lag autocorrelated constant variance errors (named AMLS) and (b) uncorrelated inhomogeneous variance errors (named HMLE). Sorooshian and Dracup (1980) reported successful testing of these measures on a two-parameter model using synthetic data. Sorooshian (1981) reported similar success with a four-parameter model. Sorooshian, Gupta, and Fulton (1983) reported an application of the two likelihood measures to the identification of the parameters of the SMA-NWSRFS model for the Leaf River Basin in Mississippi. It was reported that the HMLE measure resulted in parameter estimates much superior (in terms of conceptual realism,

sensitivity to chosen calibration data set, and forecast performance) to those obtained using either the AMLE or the SLS measures. Most recently, Lemmer and Rao (1983) reported that the HMLE measure performed very well in the estimation of the parameters of the ILLUDAS model.

Two other reports of the use of Maximum Likelihood measures of closeness are those of Goldstein and Larimore (1980) and Posada and Bras (1982). The output measurement errors were assumed Gaussian, zero mean, uncorrelated, and of constant coefficient of variation. The models calibrated were stochastic state space versions of the SMA-NWSRFS. No check on the assumption of constant coefficient of variation was reported. Also, significant correlation was detected in the identified model residuals.

Finally, an alternative approach that has been proposed for identification in the presence of correlated errors is the method of Generalized Least Squares (Kuczera, 1982; Yeh, 1982; and Williams and Yeh, 1982). Kuczera (1982) also demonstrated the usefulness of this approach in the calibration of a simple watershed model to two simultaneously correlated output sequences (streamflow and soil moisture level). These procedures, however, seem to neglect the inhomogeneous nature of the measurement error variances.

1.3.3 Selection of an Informative Data Set

Clearly, the success of any procedure for identification of the parameters of a watershed model is ultimately dependent on the nature of the quantitative data used. It has often been suggested or implied in the literature that the data used should be as "representative" of the various phenomena experienced by the watershed as possible. Many researchers have attempted to satisfy this requirement by using as large a data set as possible, without demonstratably superior results. Goldstein and Larimore (1980)

and Sorooshian, Gupta, and Fulton (1983) have noted that it is not the length of data used but the information contained in it (hydrologic variability) and the efficiency with which it is extracted that is important. Klemes (1982) has warned against the danger of "overfitting", which amounts to regarding part of the noise in the data as information (the demarkation line between information and noise often being blurred in the data). Sorooshian, Gupta, and Fulton (1983) obtained results that underscore the importance of the remarks by Klemes. They reported that, when the inhomogeneity of the error variance was accounted for in the identification procedure, the parameter estimates were much less sensitive to various characteristics of the calibration data, such as hydrologic variability and length. The use of the HMLE (see Section 1.3.2) as a measure of closeness produced conceptually realistic parameter estimates and uniformly good forecasts using only one water-year of data for identification. The use of longer data sets was found to only marginally improve the parameter estimates. They recommended that not less than one water-year of data be used to ensure adequate representation of the complete seasonal hydrologic cycle, and that wetter years are more likely to ensure adequate information about the values of the parameters.

Finally, Fiering and Kuczera (1982) recently revived some fundamental questions regarding the adequacy of streamflow measurements as the sole source of output information for model identification. They pointed to the fact that streamflow accounts for only about one-third of the throughput in a watershed, and that existing identification procedures which concentrate on the relationship between rainfall and runoff exclude two-thirds of the throughput from contributing to the identification procedure. Kuczera (1982) demonstrated that additional data such as soil moisture level

measurements can be used to improve the stability of the parameter estimates of a simple linear watershed model.

1.3.4 Algorithms for Identifying the Best Parameter Values

The algorithms most frequently used to identify the parameters of conceptual watershed models have been those belonging to the class of "direct-search" procedures, such as the Simplex method (Nelder and Mead, 1965), the Pattern Search method (Hooke and Jeeves, 1961), and the Rotating Directions method (Rosenbrock, 1960) (see, e.g., Dawdy and O'Donnell, 1965; Nash et al., 1970; Chapman, 1970; Ibbitt, 1970; Monro, 1971; Johnston and Pilgrim, 1976; Pickup, 1977; Sorooshian, 1978; Arfi, 1980; Sorooshian et al., 1981 and 1983). The reason that gradient-based techniques have seldom been used is that it is generally believed that the values of the derivatives of the model equation with respect to its parameters cannot be explicitly obtained due to the presence of threshold type parameters in the models (e.g., Johnston and Pilgrim, 1976; Moore and Clarke, 1981). Johnston and Pilgrim (1976) stated that the response surface would have discontinuous derivatives and that these would adversely affect the performance of the gradient-based optimization algorithms. Some researchers have compared the performance of direct-search algorithms to those wherein the gradients are approximated using finite difference techniques (e.g., Chapman, 1970; Ibbitt, 1970; Johnston and Pilgrim, 1976; Pickup, 1977; among others). In most cases, the direct-search procedures were found to be superior; Johnston and Pilgrim (1976) suggested that this was probably due to numerical inaccuracies arising in the numerical approximation procedures.

The only reported work which used a gradient algorithm employing explicitly computed derivatives is that of Goldstein and Larimore (1980) who

replaced the thresholds present in the SMA-NWSRFS model by smoothing functions. They employed the Levenberg-Marquardt procedure for the search, and reported good convergence in the region of the optimum. They pointed out that the Levenberg-Marquardt procedure converges very slowly if the response surface is significantly non-quadratic (as is the case with watershed models) and suggested that reparameterization in the parameter space might help. They did not, however, indicate a procedure by which this might be done and in fact suggested that, due to the complexity of watershed models, determining such a reparameterization would be almost certainly infeasible. Something along the line of optimization in a transformed parameter space was also suggested by Johnston and Pilgrim (1976) who remarked that different scalings of the parameters cause changes in the configuration of the response surface, significantly affecting the difficulty of the optimization. They remarked that progress towards the optimum could be greatly improved by rescaling the parameters with the aim of producing near-circular contours of the function (measure of closeness) being optimized, but that the form of the response surface in the multi-dimensional parameter space, and hence the best selection of transformations, was unknown.

1.3.5 Identifiability of the Model Set

As is evident from Section 1.3.1, a great deal of time and effort has been expended in trying to develop mathematical relationships that suitably describe the underlying physical phenomena occurring in the watershed. However, in order that the identification procedure has a chance of success, it is important that the model parameters be identifiable based on the quantity and quality of the information available about the watershed process and the mathematical tools at one's disposal. A review of the literature concerning

watershed modeling reveals that these considerations have been largely ignored to date. There are many reports that the parameters of watershed models are poorly identifiable using available data. One of the most often reported problems has been the inability to obtain unique "best" model parameters for any given watershed. Johnston and Pilgrim (1976) expended two years of time and effort in an unsuccessful search for a unique "best" set of Boughton model parameter values for the Lidsdale catchment in Australia. More disturbing, however, is the fact that Pickup (1977), in a synthetic data study, found that not one of four optimization algorithms he tested was able to find the "true" parameter values. Similar problems have been reported with both the Stanford Watershed Model (Ibbitt, 1970) and the SMA-NWSRFS model (Brazil and Hudlow, 1981; Goldstein and Larimore, 1980; Restrepo-Posada and Bras, 1982; and Sorooshian, Gupta, and Fulton, 1983).

One of the few works which explicitly recognized the role that model structure selection has in determining the success or failure of the identification process is that of Johnston and Pilgrim (1976). These authors noted that some of the functional forms chosen for the model subprocesses could create conditions that available parameter identification methodologies might find rather difficult to handle. In particular, they mentioned the problems caused by high levels of parameter compensation and indifference (lack of sensitivity) and the problem of discontinuities in the response surface derivatives caused by threshold parameters. However, no clear solution to any of these problems was suggested. Sorooshian and Gupta (1983) presented a discussion of the main causes of these problems. They identified the following three causes:

(a) model structure representation and nonlinearity in the parameter space;

(b) imperfect representation of the physical process by the model; and

(c) data and their associated measurement errors.

A fourth possible reason that will be discussed in this report is

(d) poorly chosen parameterizations of the structural equations of the model.

1.3.6 Evaluation of the Degree of Success of the Identification Procedure

The degree of success of an identification procedure is related to how close the identified model is to the true process. There are two aspects to this. The first is related to the conceptual realism and the precision of the parameter estimates. The second is related to the ability of the model to reproduce the behavior of the watershed. The study of parameter precision is often called parameter sensitivity analysis.

The importance of parameter sensitivity analysis in the development and use of watershed models was first pointed out by Dawdy and O'Donnell (1965). Since then, there has been only sporadic interest in the topic. Plinston (1972) introduced the idea of response surface analysis in the study of model parameter sensitivities and illustrated his discussion with examples from some simple two-parameter models. McCuen (1973) used these ideas to show how parameter sensitivity analyses can provide useful information about all phases of the watershed modeling process: model formulation, parameter identification, and verification. Mein and Brown (1978) introduced a statistical method for sensitivity analysis based on the optimal properties of the SLS function. Sorooshian (1978) suggested that a quadratic approximation to the response surface be used to study the properties of an epsilon-indifference region around the parameter set to which the search algorithm

has been converged. Sorooshian and Arfi (1982) extended the ideas of Sorooshian and introduced two indices: a measure of conditioning of the response surface called "concentricity", and a measure of "interaction" among the parameters, useful for studies of the response surface in two-dimensional subspaces. Both Sorooshian (1978) and Sorooshian and Arfi (1982) estimated the quadratic approximation by the use of finite difference approximations in the region of the optimum and reported some difficulties with this procedure. This aspect was studied by Phillips (1981), who showed that the manner in which the points were chosen for the finite difference approximation was a critical factor in obtaining a reliable estimate of the Hessian matrix. He examined a number of designs with different statistical properties and concluded that the "Minimum Bias" design gave the most reliable results in the presence of non-quadratic response surfaces. Goldstein and Larimore (1980) and Posada and Bras (1982) used the final Hessian estimate obtained during the optimization to construct an estimate of the parameter covariance matrix, based on the statistical properties of their Maximum Likelihood estimators. They used diagonal terms of the covariance matrix as indicators of the precisions of the estimates.

The ability of the identified model to reproduce the behavior of the watershed is usually tested by using the model to forecast over a data period not used in the identification procedure. The fit of the model output to the observed data should be essentially no worse than the fit during the identification data period. A review of the literature reveals that conventional statistical measures (e.g., coefficient of variation, root mean square error, percent bias, etc.) have usually served as the main criteria for testing the accuracy of calibrated models (Aitken, 1973; Clarke, 1973; Fleming, 1975). Sorooshian and Dracup (1980) and Sorooshian, Gupta, and

Fulton (1983) argued that these statistical goodness-of-fit criteria are based on the error contaminated measured values of the output and are therefore not necessarily good indicators of model fit. Kitanidis and Bras (1980) pointed out that good performance with respect to quadratic criteria does not necessarily guarantee equally satisfactory performance in forecasting the most important aspects of the hydrograph, such as the beginning of the rising limb, and the magnitude and timing of the peak. The last mentioned authors suggested the use of the "coefficient of persistence" which compares the model performance to the "no-model" prediction (predicted flow at time t + 1 equals observed flow at time t). This indicator favors models that perform well under the highly non-stationary conditions of flow forecasting. Other indicators which have been reported include the following: a linear combination of the mean ratio and its standard deviation (Cunday and Brooks, 1981); the percent root mean square error by flow group (Posada and Bras, 1982); and the percent bias by flow group (Sorooshian, Gupta, and Fulton, 1983). The last mentioned statistic was shown to be useful in exposing any tendency of the model to reproduce one aspect of the hydrograph at the expense of another.

A separate though equally important aspect of model validation that is either often ignored or goes unreported concerns the comprehensive checking (if possible) of all the assumptions that went into the model structure and parameter identification stages. Alley et al. (1980) pointed out that the lack of such checks can bring seriously into question the validity and thoroughness of the identification procedure.

Finally, Klemes (1982) has suggested that the most important characteristic of a conceptual model of the watershed process should be its credibility under diverse and varied conditions. Therefore, an objective means of

comparing the accuracy and reliability of different models would be to test their forecasting abilities during periods that are as remotely different in characteristics from the identification data period as possible.

1.4 Organization and Scope of the Research

The review of the literature presented in Section 1.3 reveals that, although 40 or more years have passed since the first simple watershed model was developed, the problem of identifying a model for a given watershed is still plagued with many problems, as well as uncertainties as to how they are to be resolved. It would appear that at least part of this problem is due to the lack of agreement and understanding on the part of researchers as to how to view the identification problem. It was partly with this in mind that the Section 1.2 discussion on the General Identification Problem was presented and used as a framework to organize the literature review.

The literature review clearly indicates that performance of watershed models is limited less by our inability to develop more complex and realistic models than it is by the inability of the mathematical procedures currently being implemented to adequately deal with the parameter estimation problem. However, it is important to note that existing watershed models may not be "identifiable" even if the most powerful and sophisticated techniques available were to be implemented. Bearing these facts in mind, the following areas have been investigated in this research.

Chapter 2 discusses the election of the model to be investigated. For this study, a simplified version of the NWSRFS model of the U.S. National Weather Service was developed. Chapter 3 discusses our investigation into the issues of model structural causes of nonidentifiability. A mathematical measure of structural identifiability is developed. The measure requires

that the model derivatives with respect to the parameters be available. To this end, a methodology for computation of model derivatives for models with threshold parameters is developed. These methods are applied to the study of the percolation equation of the NWSRFS model. The parameters of the percolation equation are shown to be nonidentifiable and a reparameterization is proposed.

In Chapter 4, we discuss the stochastic nature of the parameter identification problem and its solution by the method of Maximum Likelihood. A new likelihood procedure is proposed based on the nonlinear nature of the rating curve. The new procedure is tested using simulation examples.

The conclusions of this research are discussed in Chapter 5.

CHAPTER 2

SELECTION OF THE MODEL TO BE INVESTIGATED

2.1 The Structure of a Typical Conceptual Rainfall-Runoff Model

Most sophisticated conceptual rainfall-runoff models available today recognize the presence of different vertically stratified zones of soil in the ground (see Figure 2.1). By and large, this stratification is represented by an upper zone which extends from the surface approximately to the depth of short-rooted plants and shrubs and has a relatively high rate of lateral depletion through subsurface runoff (interflow), and a lower zone representing groundwater storage with a much slower recession rate, depleted through baseflow which leaves the watershed either through deep groundwater flow or through seepage into a surface stream or river. The lower zone is replenished through vertical percolation from the upper zone. In addition to the upper and lower soil moisture zones, there is often a surface reservoir representing interception storage, a representation for impervious area and for runoff generated from saturated soil areas. Finally, there is provision for depletion of all reservoir areas through evapotranspiration and consumption by plants. The total inflow to the stream is then added together and flow downstream is simulated by routing through a series of linear reservoirs.

The set of operations to be represented by the model, then, are:

(a) interception of rainfall;

(b) direct runoff from impervious areas;

(c) infiltration of water from surface to upper zone;

(d) surface runoff of water unable to be accepted by upper zone;

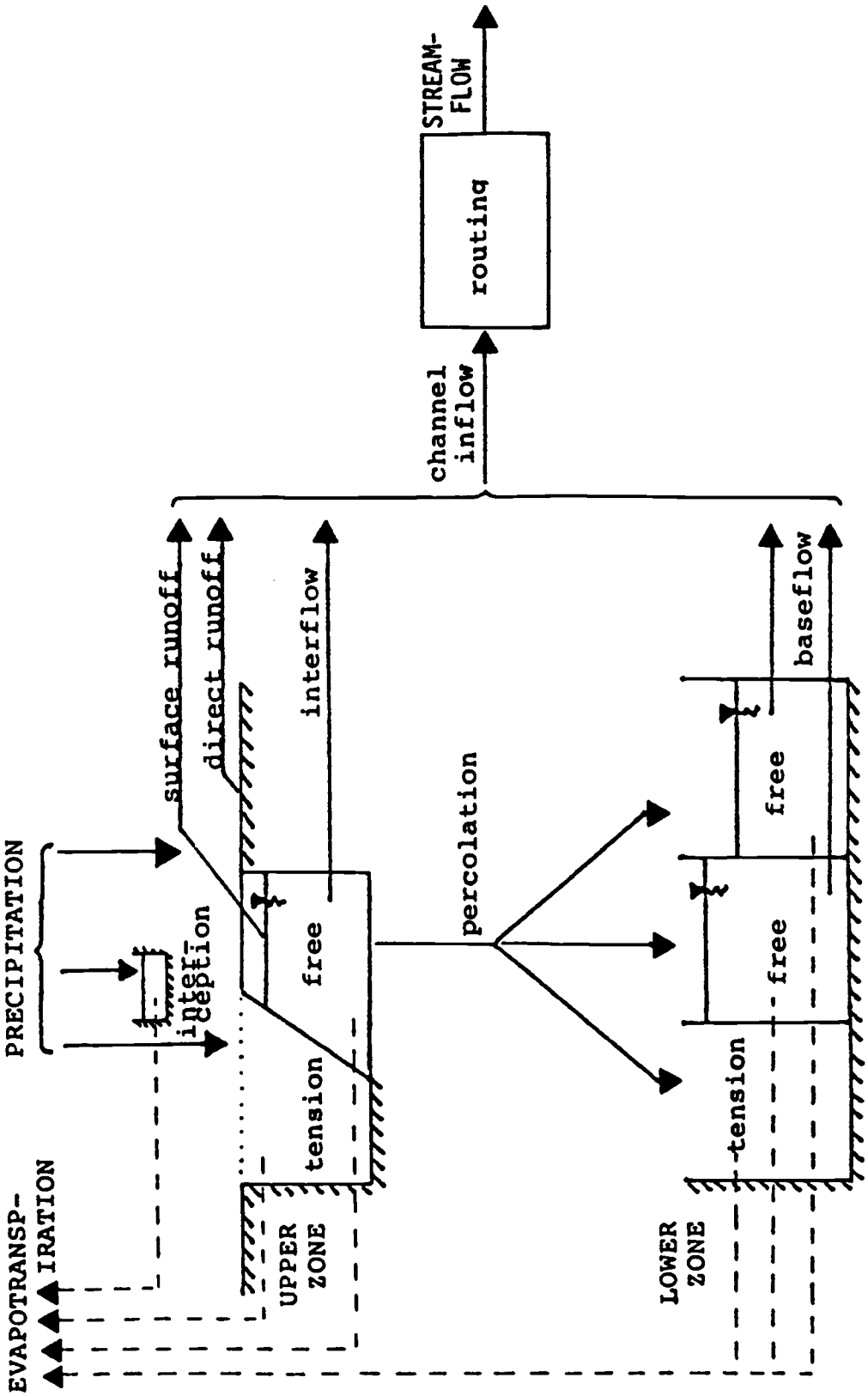


Figure 2.1. Schematic representation of a rainfall-runoff model

(e) percolation of water from upper to lower zones;

(f) interflow (lateral subsurface flow) from upper zone;

(g) baseflow (lateral groundwater outflow into river) from lower zone;

(h) evapotranspiration from all storages to the atmosphere; and

(i) routing of river inflow to downstream gauging point.

The model usually accepts lumped precipitation and potential evapotranspiration input data, which represent equivalent uniform depths of precipitation and potential evapotranspiration over the catchment area, at fixed intervals of time. The model output is usually the stream stage or flow volume, but may also consist of evaporation water loss data or volumes of subsurface water storage.

The model generally operates as follows. Precipitation input is added to the contents of the upper zone tension water storage, which is a simple reservoir limited in size by a threshold parameter. If the threshold is exceeded, the excess water enters the upper zone free water storage.

The upper zone free water storage is a tank, limited in size by a threshold parameter, depleted vertically according to a nonlinear percolation loss function, and depleted laterally by a simple proportionality rate constant. If, after percolation and lateral flow depletions have been deducted, the storage exceeds the threshold value, the excess water flows overland as runoff.

The percolation function is a nonlinear function which transfers water from the upper zone to the lower zone simulating the effects of gravity and capillary suction. The rate of transfer is a function of the lower zone moisture deficit and the upper zone contents.

Percolating water is distributed by some rule between lower zone tension and free water storages. As in the upper zone, the lower zone tension is limited in size by a threshold parameter and excess water enters the lower zone free water storage.

The free water storages are depleted laterally by simple proportionality rate constraints and are limited in size by thresholds.

The lateral outflows are added together and constitute inflows into the stream. The water is then routed downstream using a routing technique. The simplest and most commonly-used technique is a moving average transformation of the channel inflow into streamflow at the downstream gauge point.

Evapotranspiration losses from each reservoir are computed as proportions of the existing storage, first from the upper zone, and then from the lower zone if the evapotranspiration demand is not satisfied. If the demand is still not satisfied, then the remaining demand is ignored.

From the above description, it can be seen that certain major characteristics of the model are the presence of:

(a) threshold parameters;

(b) linear recession rate parameters;

(c) a nonlinear percolation function; and

(d) a linear routing function.

2.2 A Brief Description of the Soil Moisture Accounting Part of the National Weather Service's River Forecast System Model (NWSRFS)

The NWSRFS model is a highly complex mathematical representation of various aspects of physical hydrology, developed with an intent to forecasting the response of a watershed system to hydrologic inputs. Thus, it includes a catchment soil moisture accounting system, a model for the

accumulation and ablation of snow, various channel routing models, and techniques for modeling the area input of precipitation. Because of the requirement that it be applicable to any general watershed, these components are necessarily highly parameterized, and the parameters must be estimated separately for each watershed to which it is desired to apply the model.

The aspect of the NWSRFS which is of interest here is the catchment soil moisture accounting system. The soil moisture model is a version of the Sacramento Model developed and reported on by Burnash et al. (1973) (see also U.S. Department of Commerce NOAA TM NWS HYDRO 31, 1976). Like many sophisticated R-R models available today, the NWSRFS soil moisture accounting system (hereafter called SMA-NWSRFS) recognizes the presence of different vertically stratified zones of soil moisture in the ground. This stratification is represented by an upper zone which extends from the surface approximately to the depth of short-rooted plants and shrubs and has a relatively high rate of lateral depletion through subsurface runoff (interflow). A lower zone represents groundwater storage with a much slower recession rate, depleted through baseflow which leaves the watershed either through deep groundwater flow or through seepage into a surface stream or river. The lower zone is replenished through vertical percolation from the upper zone. In addition to the upper and lower soil moisture zones, there is a representation for impervious area and for runoff generated from saturated soil zones. Finally, there is provision for depletion of all reservoir areas through evapotranspiration and consumption by plants. The total inflow to the stream is then added together, and flow downstream is by routing through a series of linear reservoirs. Details of the mathematics of the model can be obtained from the NOAA technical memorandum mentioned earlier.

The SMA-NWSRFS model, exclusive of the evapotranspiration demand curve, involves 17 parameters, and therefore calibration, or "parameter estimation", is a difficult task. The National Weather Service uses a combination of "manual" and "automatic" parameter estimation techniques. Generally, a manual approach is first used to identify a reasonably good initial parameter set, and the automatic procedure is then used to refine those initial estimates. It should be noted that certain parameter values can be directly inferred or measured from various sources (e.g., topographic maps, hydrographs, etc.). However, there are other parameters, for instance the tension storage capacity magnitudes or the percolation equation parameters which can only be estimated through an estimation process. Therefore, it is extremely important that an identification procedure be established that is an non-subjective as possible and that will result in consistent parameter estimates regardless of the data being used.

2.3 The Model SIXPAR

The SMA-NWSRFS model described in Section 2.1 is a highly complex model of the land phase of the rainfall-runoff process. However, as described in the previous sections, it contains some major features common to most conceptual rainfall-runoff models, features which have been associated with problems in parameter estimation. In order to facilitate a detailed study of these major features (i.e., threshold parameter and percolation equation nonlinearities), a simpler version of the model (referred to as SIXPAR) was examined.

The model SIXPAR is a modified and simplified version of the SMA-NWSRFS model used by the U.S. National Weather Service for flood forecasting. The

model is not capable of, nor is it intended to be, a practical tool for flood forecasting. It was constructed purely for the purposes of theoretical analysis, so as to mimic the characteristic behavior and dominant modes of operation of the SMA-NWSRFS, and is therefore ideally suited for the purposes of theoretical and experimental research.

The model SIXPAR has been derived from the SMA-NWSRFS by employing the following simplifications:

(1) the tension water and lower zone supplementary water storages have been deleted;

(2) evapotranspiration is assumed negligible;

(3) channel routing is not considered; and

(4) a constant computational time interval is employed.

A schematic diagram of the model SIXPAR appears in Figure 2.2. The model preserves the significant features and dominant characteristics of the model SMA-NWSRFS. It consists of an upper zone representing subsurface water storage and a lower zone representing longer term groundwater storage, connected by vertical percolation (PERC (I), where I is the discrete time interval index). The basic structure of the nonlinear percolation equation (considered to be the heart of the SMA-NWSRFS model), the threshold nature of the reservoir operation (the dominant model nonlinearities), and the linear recession nature of the reservoir depletion rates have been preserved. The input to the model is precipitation (P(I)) and the outputs from the model are surface runoff (R(I)), subsurface flow (S(I)) and groundwater baseflow (B(I)). The three outputs add together to give channel inflow (Q(I)). The two model states are upper zone storage (US(I)) and lower zone storage (BS(I)).

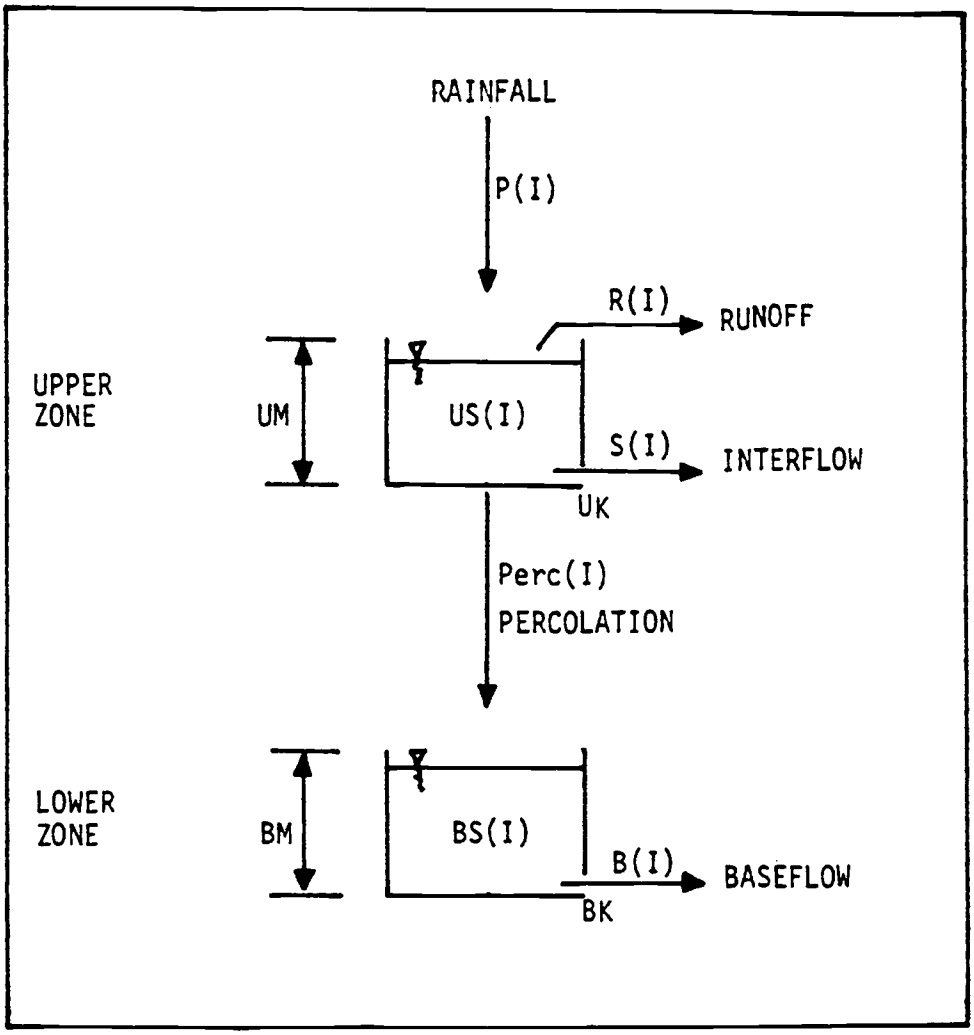


Figure 2.2. The six-parameter rainfall-runoff model SIXPAR

It is assumed that only P(I) and Q(I) are observable time series. The six parameters of the model are:

UM = upper zone maximum contents (threshold parameter), mm

BM = lower zone maximum contents (threshold parameter), mm

UK = upper zone lateral depletion constant, dimensionless

BK = lower zone lateral depletion constant, dimensionless

Z, X = parameters of the percolation equation, dimensionless

The flow chart of the model operation and the Fortran computer program for the model are included in Appendices 1 and 2.

CHAPTER 3

INVESTIGATION AND MODIFICATION OF MODEL STRUCTURE

3.1 The Issue of Identifiability

In Section 1.2, we discussed the concepts involved in identifying a conceptual model of the watershed process. This involves selection of an appropriate model set, specification of a measure of closeness, selection of an informative data set, and solving for those parameter values that optimize the measure of closeness. However, our ability to successfully solve the identification problem is hampered less by our inability to develop conceptually realistic models than by the limitations of the parameter identification tools (optimization algorithms) available to us. Hence, it is extremely important that these limitations be borne in mind while selecting the model set, i.e., one should try to ensure that the parameters of the model will be "identifiable" based on the quantity and quality of the information available and the mathematical tools at one's disposal.

Although a great many models of the watershed process have been developed (Section 1.3.1), it appears that the primary motivation behind the development of each was to construct as realistic a model as was possible. Hence, a great deal of time and effort has been expended in trying to establish mathematical relationships which accurately describe the underlying physical phenomena occurring in the watershed, while the considerations of parameter identifiability have been largely ignored. This view is supported by the many reports (Section 1.3.5) that have appeared in the literature of the inability of researchers to obtain unique "best" sets of parameters for their models. Of particular concern are the reports (see e.g., Pickup,

1977) that, even under the ideal conditions of synthetic data simulation studies, the "true" parameters of such models cannot be recovered using available techniques.

There appear to be four possible reasons why we may be unable to identify the "true" parameter values of a model in a synthetic data study. They are:

(a) The Log-Likelihood function being maximized has more than one local maximum. Hence, the nonlinear optimization algorithm may terminate at the wrong one (i.e., not at the global maximum);

(b) The optimization procedure is started in a region leading towards the global maximum but fails (terminates prematurely) due to its not being able to find one of the feasible directions of movement. This is a common occurrence when a "Type a" (direct-search) algorithm is used and long and narrow ridges are encountered on the response surface;

(c) The identification procedure is unable to distinguish between many different parameter values in the region of the optimum, i.e., the likelihood function is insensitive to perturbations of the parameters; and

(d) The global optimum of the likelihood function does not coincide with the "true" (best) parameter values. This may occur due to the random errors present in the data.

The studies reported by Ibbitt (1970), Johnston and Pilgrim (1976), Pickup (1977), Sorooshian and Dracup (1980), Sorooshian et al. (1983), and Gupta and Sorooshian (1983) seem to strongly support the notion that the inability to identify unique "best" parameter values for conceptual watershed models

is due to all of the aforementioned problems. While the first two problems have been widely recognized, the latter two do not appear to have received the attention they merit.

There appear to be two different approaches that can be employed to properly handle the problem of multiple optima. One approach is to restrict the model set to one for which the likelihood function is unimodal (e.g., model which is linear in the parameters). This may in general be an unacceptable approach as it may result in an unrealistically simple model. Also, there appear to be no general guidlines available as to how this might be done. The alternative approach is to develop a procedure that is capable of detecting the general location of all existing optima. The feasible parameter space may then be segmented into separate subspaces, each containing one optimum, whose location can be found by using the standard nonlinear optimization algorithms. Some such procedures are apparently under development but have not yet received widespread acceptance or application (e.g., Karnopp, 1963). This issue is beyond the scope of this report and will not be discussed any further here.

The second problem is related to the efficiency and reliability of optimization algorithms. The gradient-based procedures are considered to be superior in this respect to the direct-search procedures and hence the use of the former whenever possible is recommended. Furthermore, appropriate reparameteriation of the problem may often go a long way towards reducing this problem. The third problem is related to the issue of the local identifiability of the model parameters. As will be discussed in Section 3.2, identifiability is not only a function of the chosen measure of closeness (Chapter 4) and the informativeness of the data set, but is also influenced

by model structural considerations. A measure of "structural identifiability" that should prove useful in the detection of poorly-structured or parameterized models is introduced in Section 3.2.2. The application of this measure in the analysis of model parameter identifiability is discussed in Section 3.3, while Section 3.4 discusses the use of model reparameterization as a method of improving the structural identifiability of a model. Finally, Section 3.5 contains a discussion of the relationship between the concepts of structural identifiability introduced in this chapter and the concepts of parameter precision for the simple (but interesting) case of Gaussian measurement error.

3.2 Model Structural Identifiability

3.2.1 Introduction

The general concepts of identification and identifiability are basic to and have therefore received the most attention in the field of mathematical statistics. Koopmans and Reiersol (1950) report that, in 1922, the statistician R.A. Fisher distinguished as the primary group of problems in mathematical statistics the "specification of the mathematical form of the population from which the data are regarded as a sample." Hence, discussions on identification and identifiability appearing in the literature (e.g., Koopmans and Reiersol, 1950; Rothenberg, 1971) have focused on drawing inferences from the probability distribution of some observed data to an underlying model structure. This is essentially equivalent to identifying the parameters which maximize the likelihood function.

Let $\underline{\theta}^*$ be the global maximum of the likelihood function. Koopmans and Reiersol (1950) provide the following definitions.

Definition 1: The parameter point $\underline{\theta}^*$ is said to be globally identifiable if and only if there is no other $\underline{\theta}$ in the feasible space Θ that is equivalent. Two points $\underline{\theta}^1$ and $\underline{\theta}^2$ are said to be equivalent if the likelihood at $\underline{\theta}^1$ is equal to the likelihood at $\underline{\theta}^2$ ($L(\theta^1) = L(\theta^2)$).

Definition 2: The parameter point $\underline{\theta}^2$ is said to be locally identifiable if and only if there exists an open neighborhood of $\underline{\theta}^*$ containing no $\underline{\theta}$ in Θ that is equivalent.

If the likelihood function is unimodal in $\underline{\theta}$, then the two definitions are equivalent. Note that the issue of identifiability is essentially equivalent to the notion of being able to find a unique parameter value that maximizes the likelihood function.

In order to understand how nonidentifiability may arise, consider the following. A model may be defined as "a mapping M that maps the space of inputs U and the space of parameters Θ into the space of outputs Z" (i.e., $M: \Theta, U \rightarrow Z$, or $\underline{z} = M(\underline{u},\underline{\theta})$ where $\theta \in \Theta \subset R^p$, $\underline{u} \in U \subset R^n$, $\underline{z} \in Z \subset R^n$). Now the value of the likelihood function at a point $\underline{\theta}$ is obtained by running the model to get an output vector $\underline{z}$ and substituting $\underline{z}$ into the likelihood function to obtain a single number which is the measure of optimality. Diagrammatically,

$$\underline{\theta} \xrightarrow{M} \underline{z}(\underline{\theta}) \xrightarrow{L} L(\underline{z}(\underline{\theta}))$$

Note that the mapping from $\underline{\theta}$ to $L(\underline{z}(\underline{\theta}))$ involves two stages. The first is a mapping from R^p to R^n where p and n are the number of parameters and outputs, respectively. The second is a mapping from R^n to R^1. Hence, there

are two ways in which nonidentifiability might be able to arise. The first is if two distinct parameter values $\underline{\theta}^1$ and $\underline{\theta}^2$ give rise to exactly identical output vectors (i.e., $\underline{z}(\underline{\theta}^1) = (\underline{z}(\underline{\theta}^2), \underline{\theta}^1 \neq \underline{\theta}^2)$. The second is if distinct parameter values do give rise to distinct outputs but perturbations in the output vector do not give rise to changes in the value of the likelihood function. The latter issue was discussed in more detail in Chapter 2 where it was shown that this source of nonidentifiability cannot arise if $L(\underline{z})$ is an "acceptable" function (i.e., $\nabla^2_{\underline{z}} L$ is a definite matrix). Hence, the former issue is of major concern here; we can define,

> Definition 3: A model M parameterized by $\underline{\theta}$ is "structurally" identifiable if and only if M maps unique parameter values into unique model output vectors. The model M is locally structurally identifiable if and only if there exists an open neighborhood of $\underline{\theta}^*$ in which it is structurally identifiable.

Note that while the notion of identifiability advanced by Koopmans and Reiersol (1950) is related to the fact that they are dealing with random parameters, the notion presented above deals purely with the structural properties of the model, independent of whether a likelihood or any other approach is to be used for model identification. It is therefore more suitable for use in the model development stage, when the basic distributional properties of the random phenomena or even the procedure to be used for the parameter identification stage are unknown.

One other point needs to be made here. It is important to recognize that in practice we may not have to deal with models that are truly

structurally nonidentifiable so much as with models that are "nearly" structurally nonidentifiable. In other words, we may encounter a model structure wherein distinguishable parameter sets may give rise to "nearly" indistinguishable output sequences (e.g., Gupta and Sorooshian, 1983). In such cases, numerical inaccuracies and computer round-off error may give rise to computational difficulties and ambiguities in the identification procedure. Hence, it is extremely important that the model set be selected so as to be properly structurally identifiable.

3.2.2 A Measure of Structural Identifiability

The task of checking for structural identifiability of a model set using the definition is clearly far beyond the capabilities of any human or even a computer since an infinity of possible parameter-output combinations exists for any given input sequence. Testing for global structural identifiability is therefore in most cases practically impossible. We can, however, construct measures that test for local structural identifiability of a model in the region of a parameter point of interest.

Let $\{z_t(\underline{\theta}),\ t = 1, \ldots n\}$ be the sequence of model outputs associated with parameter set $\underline{\theta}$ and let $\{z_t(\theta) + \Delta\underline{\theta}),\ t = 1, \ldots n\}$ be a sequence of outputs associated with a new parameter set $\underline{\theta} + \Delta\underline{\theta}$ obtained by a small perturbation $\Delta\underline{\theta}$ of the parameters. In order that the model M be structurally identifiable at $\underline{\theta}$, we require that $\{z_t(\underline{\theta}+ \Delta\underline{\theta}),\ t = 1, \ldots n\}$ be distinguishable from $\{z_t(\theta),\ t = 1, \ldots n\}$ for all $\Delta\underline{\theta} \neq 0$ such that $\underline{\theta} + \Delta\underline{\theta} \in \Theta$. For this to be true, we require that $\Delta z_t = z_t(\underline{\theta} + \Delta\underline{\theta}) - z_t(\underline{\theta}) \neq 0$ for at least one $t = \{1, \ldots n\}$, provided $\Delta\underline{\theta} \neq 0$. An appropriate measure of the differences in the two output sequences is given by the "acceptable" function (see Section 4.1),

$$SI(\underline{\theta},\underline{\Delta\theta}) = \sum_{t=1}^{n} \{z_t(\underline{\theta}+ \underline{\Delta\theta}) - z_t(\underline{\theta})\}^2 \tag{3.1}$$

$$= \sum_{t=1}^{n} \{\Delta z_t(\underline{\theta}, \underline{\Delta\theta})\}^2 \tag{3.2}$$

Note that $SI(\underline{\theta}, \underline{\Delta\theta}) = 0$ when $\underline{\Delta\theta} = 0$ and is greater than or equal to zero for all $\underline{\Delta\theta} \neq 0$. Clearly, in order that the model be structurally identifiable at $\underline{\theta}$, we require that $SI(\underline{\theta}, \underline{\Delta\theta}) \neq 0 \; \forall \; \underline{\Delta\theta} \neq 0$. A sufficient condition to guarantee this is that the function $SI(\underline{\theta}, \underline{\Delta\theta})$ be strictly convex over all $\underline{\Delta\theta}$. This implies that the matrix of first partial derivatives of $SI(\underline{\theta}, \underline{\Delta\theta})$ with respect to $\Delta\theta$ be of full rank, i.e.,

$$\text{Rank } \nabla_{\Delta\theta} SI(\theta, \Delta\theta) = p \tag{3.3}$$

or equivalently that the Hessian matrix of $SI(\underline{\theta}, \underline{\Delta\theta})$ with respect to $\underline{\Delta\theta}$ be positive definite, i.e.,

$$\nabla^2_{\Delta\theta} SI(\underline{\theta}, \underline{\Delta\theta}) = \frac{d^2SI(\theta, \Delta\theta)}{d\Delta\theta_i \, d\Delta\theta_j} \quad (i,j = 1, \ldots p) > 0 \tag{3.4}$$

It is simple to show, using Equation (3.1), that

$$\{\nabla^2_{\Delta\theta} SI\}_{ij} = 2 \sum_{t=1}^{n} \left\{ \frac{dz_t(\underline{\theta} + \underline{\Delta\theta})}{d\Delta\theta_i} \quad \frac{dz_t(\underline{\theta} + \underline{\Delta\theta})}{d\Delta\theta_j} \right.$$

$$\left. + \; 2 \sum_{t=1}^{n} \{\Delta z_t(\underline{\theta}, \underline{\Delta\theta}) \frac{d^2 z_t(\underline{\theta} + \underline{\Delta\theta})}{d\Delta\theta_i \; d\Delta\theta_j} \right\} \tag{3.5}$$

In order to test for structural identifiability, we must evaluate $\nabla^2_{\Delta\theta} SI$ and and check if it is positive definite for all $\underline{\theta}$ and $\underline{\Delta\theta}$. In practice, this is

rather difficult except in the case of extremely simple problems such as a linear model where $\nabla^2_{\Delta\theta}SI$ is a constant for all $\underline{\theta}$ and $\underline{\Delta\theta}$. If we solve, however, for the limit of $\nabla^2_{\Delta\theta}SI$ as $\underline{\Delta\theta} \longrightarrow 0$, we will obtain a measure of the structural identifiability in the region local to $\underline{\theta}$, i.e.,

$$\lim_{\underline{\Delta\theta} \longrightarrow 0} \{\nabla^2_{\theta}SI\}_{ij} = 2 \sum_{t=1}^{n} \left\{ \frac{dz_t(\underline{\theta})}{d\theta_i} \quad \frac{dz_t(\underline{\theta})}{d\theta_j} \right\} \tag{3.6}$$

or equivalently,

$$\nabla^2_{\theta}SI(\underline{\theta}) = \lim_{\underline{\Delta\theta} \longrightarrow 0} \nabla^2_{\theta}SI(\underline{\theta}\ \underline{\Delta\theta}) = 2\nabla_{\theta}\underline{z}(\underline{\theta}) \cdot \nabla_{\theta}\underline{z}(\underline{\epsilon})^T \tag{3.7}$$

where $\nabla^2_{\theta}SI(\underline{\theta})$ will be called the structural identifiability matrix at $\underline{\theta}$. The structural identifiability matrix can be either positive definite or positive semidefinite (note from Equation (3.6) that the diagonal elements of $\nabla^2_{\theta}SI(\underline{\theta})$ are all non-negative). If the matrix is positive definite, then the model is locally structurally identifiable. If it is positive semidefinite, then the model is structurally nonidentifiable. Many tests are available to detect the positive semidefiniteness of a matrix. Some of these are:

(a) Rank of the matrix is not full;

(b) Determinant of the matrix is zero;

(c) The matrix has some zero eigenvalues.

3.3 Parameter Sensitivity-Identifiability Analysis

3.3.1 Introduction

In practice, we will rarely, if ever, encounter an exactly semidefinite matrix while using a computer, due to round-off error, etc., or due to the

fact that the model is not quite nonidentifiable. In the latter case, we might call the model nearly nonidentifiable (we shall discuss what we mean by this in a few lines). Furthermore, we would like to know which parameters or parameter combinations the model output is insensitive to and are hence responsible for the nonidentifiability problem. This latter aspect is related to what is often called a parameter sensitivity analysis.

3.3.2 The Region of Indifference

The structural identifiability matrix (Equation 3.4) is the second derivative matrix of the function $SI(\underline{\theta}, \underline{\Delta\theta})$ given in Equation (3.1) which was chosen as the measure of the deviation between two model output sequences. The function $SI(\theta, \Delta\theta)$ can be expanded to second order of a Taylor series about the point $\Delta\theta = 0$ to give:

$$SI(\underline{\theta},\underline{\Delta\theta}) \simeq SI(\underline{\theta},0) = \underline{\Delta\theta}^T \cdot \nabla_{\Delta\theta} SI(\underline{\theta},0) = \frac{1}{2}\, \underline{\Delta\theta}^T \nabla^2_{\Delta\theta} SI(\underline{\theta},0)\, \underline{\Delta\theta} \qquad (3.8)$$

and since $SI(\underline{\theta},0)$ and $\nabla_{\Delta\theta} SI(\underline{\theta},0)$ are equal to zero,

$$SI(\underline{\theta},\underline{\Delta\theta}) \simeq \frac{1}{2}\, \underline{\Delta\theta}^T \cdot \nabla^2_{\Delta\theta}\, SI(\underline{\theta}) \cdot \underline{\Delta\theta} \qquad (3.9)$$

Note that if the model M is linear in the parameters, then exact equality holds, while in general the approximation is valid locally in some neighborhood of the parameter value $\underline{\theta}$. Now $SI(\underline{\theta},\underline{\Delta\theta})$ measures the difference between the model outputs at $\underline{\theta} + \underline{\Delta\theta}$ and $\underline{\theta}$ (note $SI(\underline{\theta},\underline{\Delta\theta}) > 0$). Let $\psi/2$ represent the difference in value of $SI(\underline{\theta}, \underline{\Delta\theta})$ for which the two model output sequences are, for practical purposes, indistinguishable. Note also that Equation

(3.9) is the equation of a hyperelliptical surface in the parameter space (centered at $\underline{\theta}$), for a fixed value of $SI(\underline{\theta}, \underline{\Delta\theta})$. Hence, the equation:

$$\Delta\theta^T \cdot \nabla^2_\theta SI(\underline{\theta}) \cdot \underline{\Delta\theta} < \psi \qquad (3.10)$$

represents the approximate region in the parameter space around $\underline{\theta}$ for which the model output sequences are considered to be indistinguishable. We shall call this the region of near-nonidentifiability, or the indifference region.

3.3.3 Parameter Identifiability Measure

A two-parameter example of a hyperellipsoid described by Equation (3.10) is shown in Figure 3.1. The size and orientation of this ellipsoid describe the region of nonidentifiability. Therefore, it is desirable to quantify some of these geometric notions into computable indices that indicate various properties of the region that are of interest to us. The most important of these properties, in the context of this paper, is the degree to which nonidentifiability is related to the compensating effects of the simultaneous variation of two or more parameters on the model output. Sorooshian and Arfi (1982) proposed two indices named "Concentricity" and "Interaction" to study this phenomenon (see Appendix 3). These indices, however, only measure interdependence in two-parameter subspaces. An alternative approach involves computing the normalized inverse of the structural identifiability matrix (in a manner akin to computing the correlation matrix from the parameter covariance) so that the diagonal elements equal 1.0 and the off-diagonal elements (< 1.0) indicate the degree of parameter interdependence. While the latter approach provides information about interdependence in the multiparameter space, such information is typically difficult to extract and interpret.

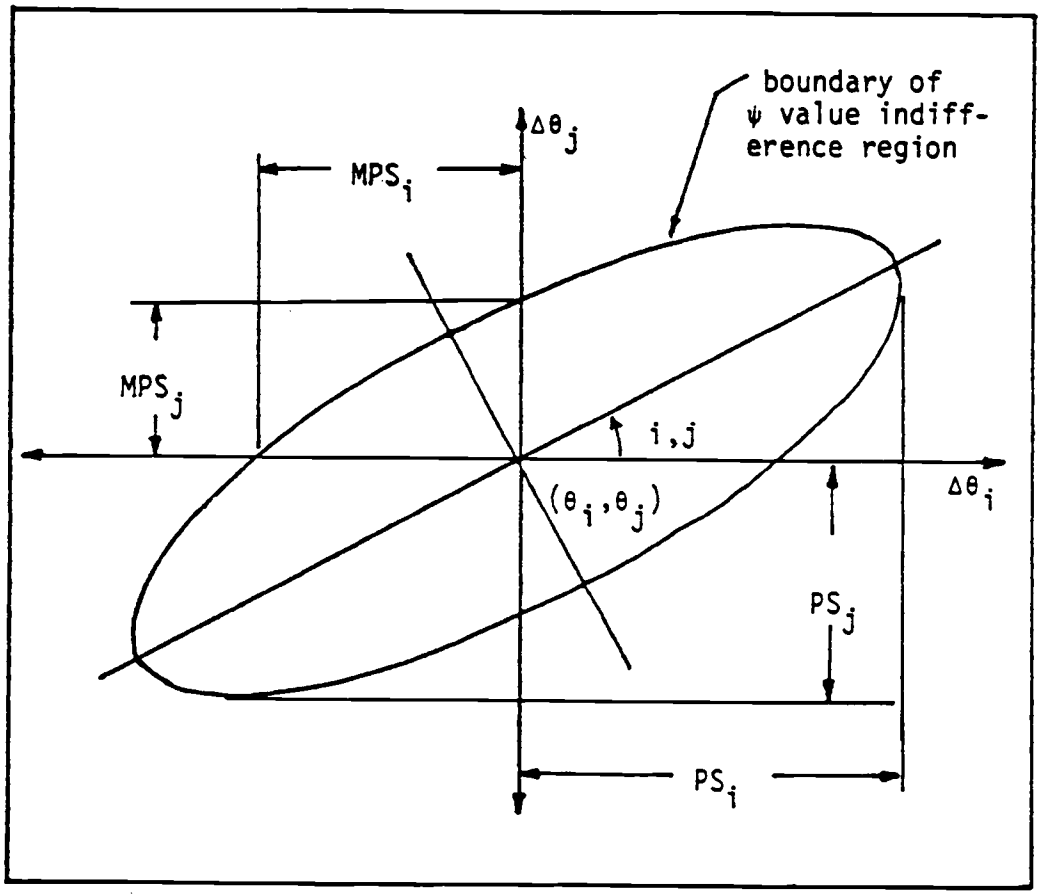


Figure 3.1. Two-parameter example of an indifference region

Our experience with parameter sensitivity analysis methodologies available in the literature (see Appendix 3 for a description of some of these methods) is that they are limited in their abilities to describe adequately the nature of parameter interrelationships when the dimension of the parameter space is greater than 2 or 3. Therefore, we suggest a measure called the "Sensitivity Ratio", which is designed to provide useful information about interactions among many parameters simultaneously in the multiparameter space. It is useful specifically for isolating those parameters that are highly compensating and poorly identifiable.

Consider Figure 3.1, which may serve as an illustrative two-parameter example. Imagine for a moment that the major and minor axes of the ellipsoid describing the indifference region were oriented along the parameter axis directions. ($\nabla^2_{\theta} SI(\underline{\theta})$ is a diagonal matrix). In this case, the maximum that each parameter θ_i could vary (allowing other parameters to vary freely) while remaining within the indifference region would be given by its conditional parameter sensitivity $CPS_i(\underline{\theta})$. In general, however, the ellipsoid will be oriented in some other manner such as in Figure 3.1 ($\nabla^2_{\theta} SI(\underline{\theta})$ is not a diagonal matrix), indicating interdependence and compensation effects among parameters. In these cases, if all parameters are free to vary simultaneously, parameter θ_i would be capable of moving a maximum distance given (Figure 3.1) by $PS_i(\underline{\theta})$ (parameter sensitivity index). It is shown in Appendix 3 that

$$CPS_i(\underline{\theta}) = \pm \left[\frac{\psi}{s_{ii}}\right]^{1/2} \tag{3.11}$$

and in Appendix 4 that $PS_i(\underline{\theta})$ is (in the p-dimensional parameter space)

$$PS_i(\underline{\theta}) = \pm \left[\frac{\Psi}{(s_{ii} - \underline{g}_i^T \cdot G_i^{-1} \cdot \underline{g}_i)}\right]^{1/2} \qquad (3.12)$$

where

G_i = the (p-1) x (p-1) submatrix of $\nabla^2_\theta SI(\underline{\theta})$ obtained by deleting the i^{th} row and column;

$\underline{g}_i$ = the (p-1) x 1 vector equivalent to the i^{th} column of $\nabla^2_\theta SI(\underline{\theta})$ with i^{th} element deleted; and

s_{ij} = the ij^{th} element of $\nabla^2_\theta SI(\underline{\theta})$.

Note the similarity in form of $CPS_i(\underline{\theta})$ (Equation 3.11) and $PS_i(\underline{\theta})$ (Equation 3.12). It should be noted that since $\nabla^2_\theta SI(\underline{\theta})$ is a non-negative definite matrix, so is G_i (for all i=1, ... p) and hence the denominator in Equation (3.12) is always less than or equal to s_{ii} (implying that $PS_i(\underline{\theta}) \geq CPS_i(\underline{\theta})$). By taking the ratio of $PS_i(\underline{\theta})$ to $CPS_i(\underline{\theta})$, we get a nondimensional measure of the amount that other model parameters (θ_j, $j \neq i$) compensate for the changes in model output caused by perturbations of parameter θ_i. We call this the Sensitivity Ratio (η_i);

$$\eta_i = \frac{PS_i(\underline{\theta})}{CPS_i(\underline{\theta})} = \left[\frac{s_{ii}}{s_{ii} - \underline{g}_i^T \cdot G_i^{-1} \cdot \underline{g}_i}\right]^{1/2} \qquad (3.13)$$

When $\eta_i = 1$, there is no compensation for the effects of parameter θ_i on the model output by the other parameters. As η_i gets larger, this indicates poorer and poorer identifiability of parameter θ_i in relation to the other model parameters.

3.3.4 Parameter Sensitivity and Precision

There is a strong relationship between the study of parameter sensitivity using concepts of model structural identifiability and the study of parameter precision, which is associated with the stochastic properties of the data. This relationship is examined here; the analysis will focus mainly on the simple case of independent, homoscedastic, Gaussian output error and, in places, involves simplifying assumptions.

Consider the case where the errors in the output observations are independently distributed as Gaussian with zero mean and constant variance equal to σ^2. The log-likelihood function (see, e.g., Bard, 1974) for this case is given by:

$$L(\underline{\theta}) = -\frac{n}{2} \text{Ln}(2\pi) - \frac{n}{2} \text{Ln}(\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=1}^{n} (\tilde{z}_t - z_t(\underline{\theta}))^2 \qquad (3.14)$$

where

$\tilde{z}_t$ = observed streamflow at time t;
n = number of observations; and
σ^2 = variance of the observation error.

Let $\hat{\underline{\theta}}$ denote the maximum likelihood values of the parameters. The Hessian matrix for this case at $\hat{\underline{\theta}}$ is given by:

$$\{\nabla_\theta L(\hat{\underline{\theta}})\}_{ij} = -\frac{1}{\sigma^2} \sum_{t=1}^{n} \left[\frac{\partial z_t(\hat{\underline{\theta}})}{\partial \theta_i} \frac{\partial z_t(\hat{\underline{\theta}})}{\partial \theta_j} + (\tilde{z}_t - z_t(\hat{\underline{\theta}})) \frac{\partial^2 z_t(\hat{\underline{\theta}})}{\partial \theta_i \partial \theta_j} \right] \qquad (3.15)$$

Hence,

$$\nabla_\theta^2 L(\hat{\underline{\theta}}) = -\frac{1}{2\sigma^2} \{\nabla_\theta^2 SI(\hat{\underline{\theta}}) + D_\theta(\hat{\underline{\theta}})\} \qquad (3.16)$$

where

$$\{D_\theta(\hat{\underline{\theta}})\}_{ij} = 2\sum_{t=1}^{n}(\tilde{z}_t - z_t(\hat{\underline{\theta}}))\frac{\partial^2 z_t(\hat{\underline{\theta}})}{\partial\theta_i\,\partial\theta_j} \tag{3.17}$$

The covariance matrix (Σ_Θ) of the parameter estimates is given by the inverse of the expected value of $[-\nabla^2_\theta L(\hat{\underline{\theta}})]$. Hence,

$$\Sigma_\theta = E\{-\nabla^2_\theta L(\hat{\theta})\}^{-1} = 2\sigma^2\,[\nabla^2_\theta SI(\hat{\theta}) + E\{D_\theta(\hat{\underline{\theta}})\}]^{-1} \tag{3.18}$$

If the model is an exact representation of the true physical process (as in a synthetic model study), then $E\{\tilde{z}_t\} = z_t(\hat{\underline{\theta}})$. On the other hand, if the model is approximately linear (locally), then $(\partial^2 z_t(\hat{\underline{\theta}})/\partial\theta_i\,\partial\theta_j)$ will be zero. In either of these cases, $E\{D_\theta(\hat{\underline{\theta}})\}$ will be equal to zero. In such a situation, we have (for nonsingular $\nabla^2_\theta SI(\hat{\underline{\theta}})$):

$$\Sigma^1_\theta = 2\sigma^2 \cdot \{\nabla^2_\theta SI\,(\hat{\underline{\theta}})\}^{-1} \tag{3.19}$$

In general, however, we have $E\{\tilde{z}_t\} = z_t \neq z_t(\hat{\underline{\theta}})$, where z_t is the true value of the output, and hence,

$$\Sigma_\theta = 2\sigma^2\,[\nabla^2_\theta SI\,(\hat{\underline{\theta}}) + \bar{D}_\theta(\hat{\underline{\theta}})]^{-1} \tag{3.20}$$

where

$$\bar{D}_\theta(\hat{\underline{\theta}}) = E\{\bar{D}_\theta(\hat{\underline{\theta}})\} \tag{3.21}$$

Using the matrix inversion lemma (Kailath, 1980) if ${}_{\theta}\nabla^2 SI(\hat{\underline{\theta}})$ and $\bar{D}_{\theta}(\hat{\underline{\theta}})$ are non-singular positive definite,

$$\Sigma_{\theta} = 2\sigma^2 \left(\nabla^2_{\theta} SI^{-1} - \nabla^2_{\theta} SI \left[\nabla^2_{\theta} SI^{-1} + \bar{D}_{\theta}^{-1}\right]^{-1} \nabla^2_{\theta} SI\right)^{-1} \tag{3.22}$$

where, for readability, the dependence on $\hat{\underline{\theta}}$ has been omitted, and hence we have that:

$$\Sigma_{\theta} \leq \Sigma^1_{\theta} \tag{3.23}$$

Hence Σ^1_{θ} is a conservative estimate of the value of the covariance of the parameter estimates. It is related to the structural identifiability matrix by Equation (3.19).

If Equation (3.19) holds approximately, the measures of parameter precision and parameter identifiability are related intimately in that the precision of a parameter estimate is directly proportional to the product of its parameter sensitivity index and the output error standard deviation. This implies that, for the more sensitive model parameters ($PS_j(\underline{\theta})$ is smaller), parameter precision is much less affected by larger output measurement errors. This supports an intuitive notion that a well constructed model is one whose output is relatively sensitive to parameter variations.

The above analysis is for the case of Guassian, independent, homoscedastic error. For the case of a more general likelihood function [$L(\underline{z}(\underline{\theta})$], the relationship between parameter precision and identifiability is not quite as simple. In the general case, the relationship can be computed using:

$$\nabla_\theta^2 L(\hat{\theta}) = \nabla_\theta \underline{z}^T(\hat{\underline{\theta}}) \cdot \nabla_z^2 L(\hat{\underline{\theta}}) \cdot \nabla_\theta \underline{z}(\hat{\underline{\theta}}) + G(\hat{\underline{\theta}}) \tag{3.24}$$

where

$$\{G\}_{ij} = \left[\frac{\partial^2 z_1}{\partial\theta_i\,\partial\theta_j}, \ldots, \frac{\partial^2 z_n}{\partial\theta_i\,\partial\theta_j} \right] \cdot \nabla_z L \tag{3.25}$$

3.4 Evaluation of the Derivatives

3.4.1 Introduction

In order to compute the sensitivity measure discussed in Section 3.1, we require the derivatives of the model output with respect to the parameters. These derivatives are also required if "gradient"-type search algorithms are to be implemented for parameter estimation. As discussed in Section 1.3.4, it has been generally believed that the derivatives could not be computed or were too difficult to compute. In this section, we discuss this issue and outline a theory for practical computation of the derivatives.

3.4.2 Modality in the Behavior of CRR Models

A typical rainfall-runoff catchment model computes the response to precipitation input in two stages. In the first stage, the precipitation input is used to augment the state of the system (i.e., to update the quantities of moisture stored in the various zones of the surface and subsurface land mass). In the second stage, the output of the model is computed as a function of the present updated state of the system. Thus, the model belongs to the general class of so-called "state-space" models. Assuming that model computations are made at discrete times, it may be expressed mathematically (in general form) in terms of the vector equations:

$$\underline{X}_t = A\,(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) \quad \text{-- State Equation} \tag{3.26}$$

$$\underline{Z}_t = B\,(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) \quad \text{-- Output Equation} \tag{3.27}$$

where

$\underline{X}_t$ = vector of model states at time t (e.g., depth of groundwater storage);

$\underline{U}_t$ = vector of model input values at time t (e.g., precipitation depths);

$\underline{Z}_t$ = vector of model output values at time t (e.g., streamflow volumes or rates); and

$\underline{\theta}$ = vector of model parameters.

The functional forms of $A(\cdot)$ and $B(\cdot)$ and the values of the parameters $\underline{\theta}$ determine the behavioral characteristics of the model. If $A(\cdot)$ and $B(\cdot)$ are linear functions of the states and inputs, then unit changes in the input cause proportional changes in the output. If, however, the functions $A(\cdot)$ and $B(\cdot)$ are such that the output is not a linear function of the input, then the behavior of the system cannot be predicted in this simple way. Although many models have been conceptualized in a state-space framework, few have so far been treated mathematically as such (see, e.g., the work of Kitandis and Bras (1980a,b) and Restrepo-Posada and Bras (1982) with the U.S. National Weather Service's SWA-NWSRFS model).

One of the peculiarities of a CRR catchment model is that it contains what are (in the hydrologic literature) termed "threshold" parameters. The practical effect of these parameters is to cause the model to operate in different modes of behavior, depending on its state. As an example, consider the simple discrete time moisture depth storage reservoir depicted in Figure 3.2, which is a common element in many catchment models. Depletion (S_t) of the reservoir from below occurs at a rate K (unit: time

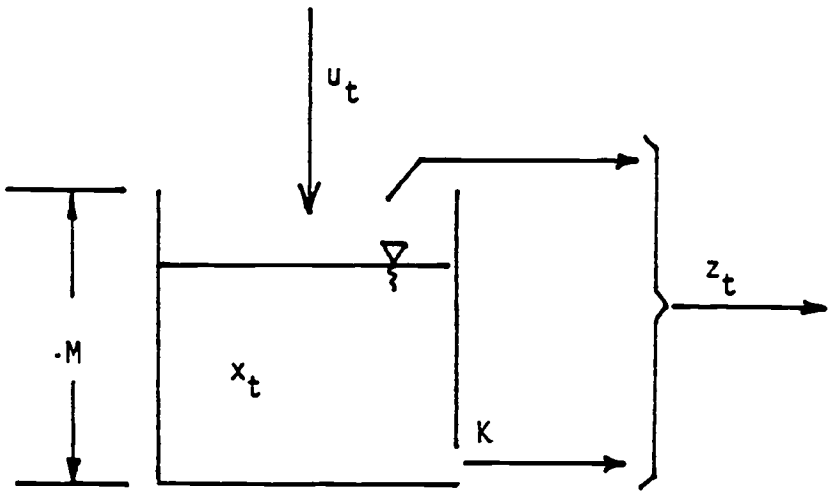


Figure 3.2. Simple single reservoir model

inverse, T^{-1}) proportional to its contents. Its capacity is limited to a maximum value M (unit: length, L). When the precipitation input is such that the storage capacity is exceeded, the excess moisture (R_t) flows over the top of the reservoir (i.e., total outflow consists of both depletion from below, S_t, and excess moisture spillover, R_t). In this simple reservoir model, the two parameters are K and M. Notice that the model operates in two modes. Let:

x_{t-1} = state of model at beginning of time interval t; and

u_t = precipitation input during time interval t.

and

$$\tilde{x}_t = x_{t-1} + u_t \tag{3.28}$$

$\tilde{x}_t$ represents the intermediate model state, i.e., the state of the model increased by the amount of input into the reservoir, and prior to the outputs being computed and depleted from the reservoir. When $\tilde{x}_t \leq M$, the model the model operates in mode one and the equations are:

$$x_t = (1-K)(x_{t-1} + u_t) \tag{3.29}$$

$$z_t = S_t = K\,(x_{t-1} + u_t) \tag{3.30}$$

when $\tilde{x}_t > M$, the model operates in mode two and the equations are:

$$x_t = (1-K)M \tag{3.31}$$

$$z_t = S_t + R_t = x_{t-1} + u_t - (1-K)M \tag{3.32}$$

The source of this modality of behavior is the "threshold parameter" M. The model is clearly linear in each mode ($z_t \propto u_t$), but the proportionality constant is equal to K in mode one while it is equal to 1 in mode two. Mode two represents system behavior under conditions of "saturation".

3.4.3 Mathematical Representation of a Modal-Type Model

A mathematical model of a dynamic process is typically represented in terms of a symbolic flow chart before it is written out in one or another computer language and programmed into a computer. A typical (hypothetical) flow chart is presented in Figure 3.3. It represents the sequence of operations of a model during a single computational time interval. Each rectangular box represents a model suboperation C_i, i=1, ..., 5, (e.g., moisture transfers into, out of, or between soil moisture storages) and each diamond-shaped box containing the word IF represents a point of branching in the program. Each such point of branching is associated with a threshold element in the model (for a specific example, see Figure 3.5).

In Figure 3.4, the flow chart has been decomposed and written out by tracing the possible routes from the top of the chart to the bottom. Each feasible route represents a possible mode of operation. Having obtained the various possible modes, say 'q' in number, the functional relations for each mode in terms of the state and output equations can be written out, as in the simple example presented earlier. Thus, we end up with q sets of equations, one for each mode, each set explicitly representing the model state and output for that time step in terms of the model parameters, the state at the end of the previous time step and input during the time step. The equations can be represented by:

$$\underline{X}_t = \begin{cases} A_1(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) \text{ if in mode one during time interval } t \\ \vdots \\ A_q(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) \text{ if in mode q during time interval } t \end{cases} \tag{3.33}$$

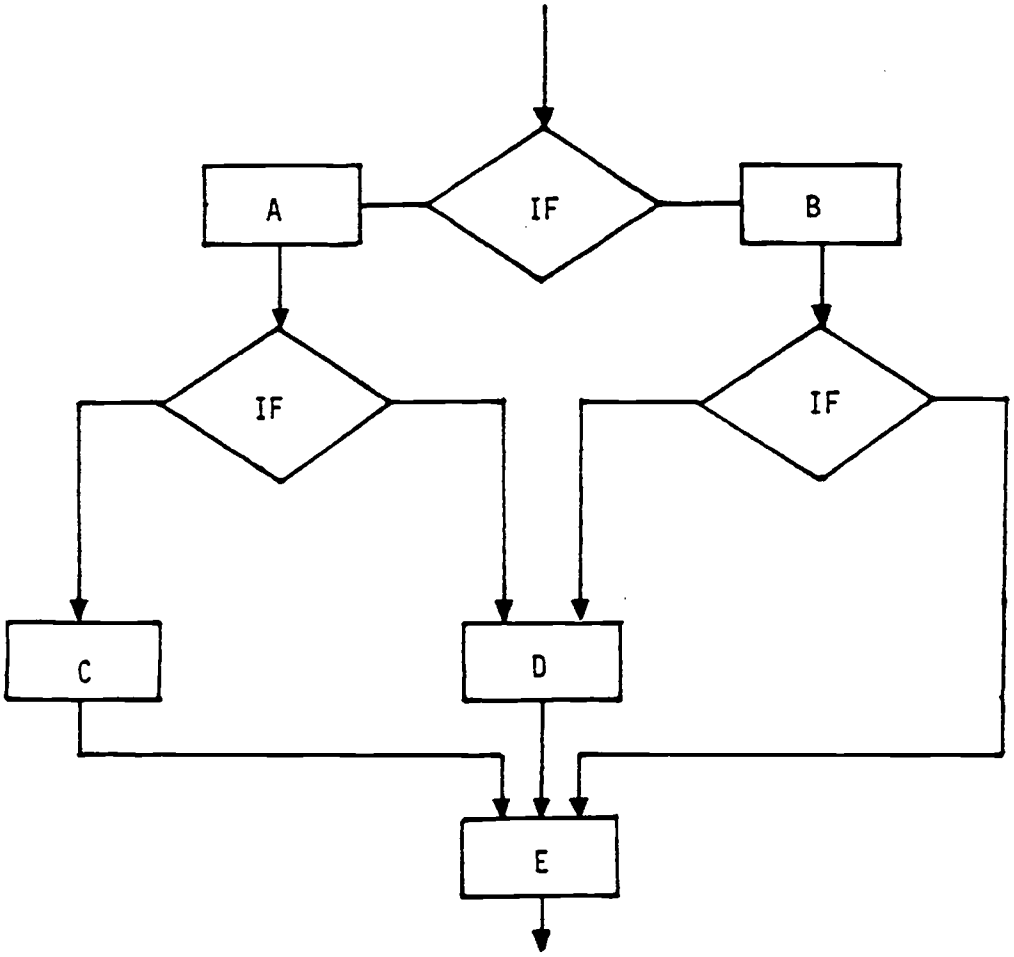


Figure 3.3. Typical computer program flowchart for a model with threshold parameters

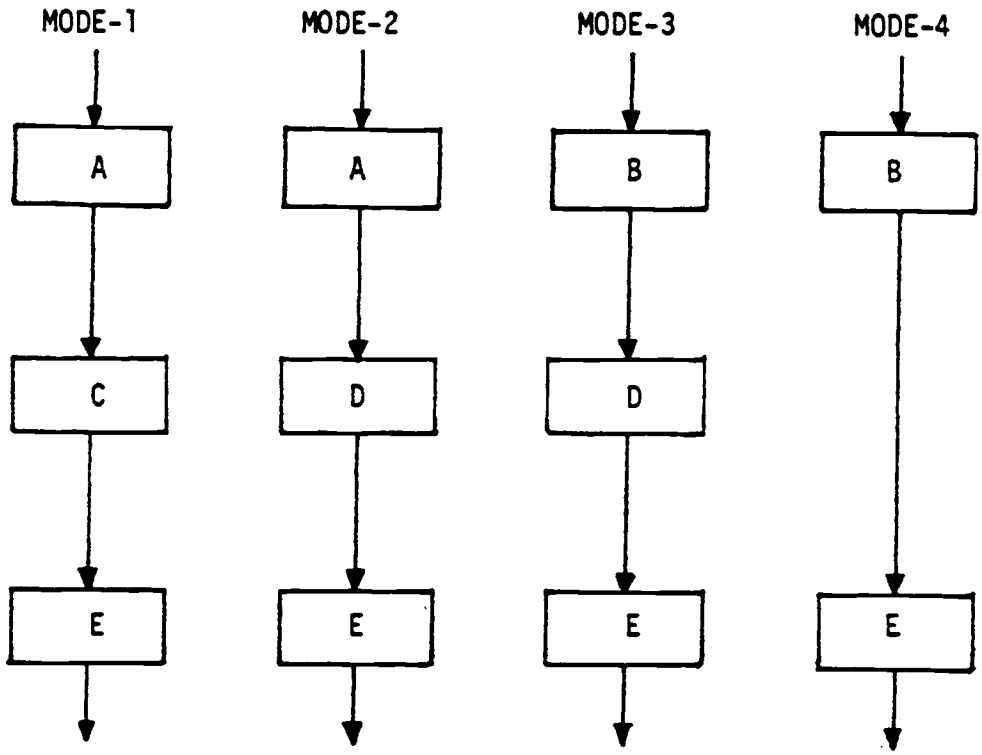


Figure 3.4. Modal decomposition of the flowchart in Figure 3.3.

$$\underline{Z}_t = \begin{cases} B_1(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) \text{ if in mode one during time interval } t \\ \vdots \\ B_q(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) \text{ if in mode q during time interval } t \end{cases} \tag{3.34}$$

A more compact way to write these equations is

$$\underline{X}_t = \sum_{j=1}^{q} T_j(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) \cdot A_j(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) \tag{3.35}$$

$$\underline{Z}_t = \sum_{j=1}^{q} T_j(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) \cdot B_j(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) \tag{3.36}$$

where

$$T_j(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) = \begin{cases} 1 \text{ if in mode j during time interval } t \\ \\ 0 \text{ if not in mode j during time interval } t \end{cases} \tag{3.37}$$

and

$$\sum_{j=1}^{q} T_j(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t) = 1 \tag{3.38}$$

Note that the above mathematical representation introduces a new quantity, the threshold function $T_j(\cdot)$ which takes on a binary value of either zero or one, depending on the values of $\underline{\theta}$, $\underline{X}_{t-1}$ and $\underline{U}_t$. The summation over T_j equal to one indicates that only one system mode can be active at a given time step.

3.4.4 The Derivative Equations

The model equations (3.33-3.38) are now in a form amenable to differentiation. For simplicity we shall denote $A_j(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t)$, $B_j(\underline{\theta}, \underline{X}_{t-1}, \underline{U}_t)$

and $T_j(\_, \underline{X}_{t-1}, \underline{U}_t)$ by the short forms A_{jt}, B_{jt}, and T_{jt}, respectively. Noting that the model equations are continuous in the state and parameter spaces, we must show in Appendix 9 that the expressions for the first derivatives are:

$$\frac{\partial \underline{X}_t}{\partial \theta_i} = \sum_{j=1}^{q} T_{jt} \left[\frac{\partial A_{jt}}{\partial \theta_i} + \frac{\partial A_{jt}}{\partial \underline{X}_{t-1}} \cdot \frac{\partial \underline{X}_{t-1}}{\partial \theta_i} \right] \qquad \forall i \qquad (3.39)$$

$$\frac{\partial \underline{Z}_t}{\partial \theta_i} = \sum_{j=1}^{q} T_{jt} \left[\frac{\partial B_{jt}}{\partial \theta_i} + \frac{\partial B_{jt}}{\partial \underline{X}_{t-1}} \cdot \frac{\partial \underline{X}_{t-1}}{\partial \theta_i} \right] \qquad \forall i \qquad (3.40)$$

where, in keeping with the manner in which conceptual rainfall-runoff models are constructed, the threshold functions are assumed to be continuous either from the left or from the right (see Appendix 9). All of the components of (3.39) and (3.40) (and hence the derivatives) can be calculated.

3.5 Analysis of a Linear Reservoir Model with Threshold

3.5.1 Introduction

In this section, we illustrate the usefulness of the theory presented above, with a discussion on the identifiability of threshold-type parameters such as are commonly found in CRR models. The discussion serves to highlight the important influence that the characteristics of the data used for calibration have on model structural identifiability.

3.5.2 Theory

Consider the simple single reservoir model of Figure 3.2. The flow chart for this model is depicted in Figure 3.5. The modal representation of the model is given by (3.29)-(3.32). The threshold functions are:

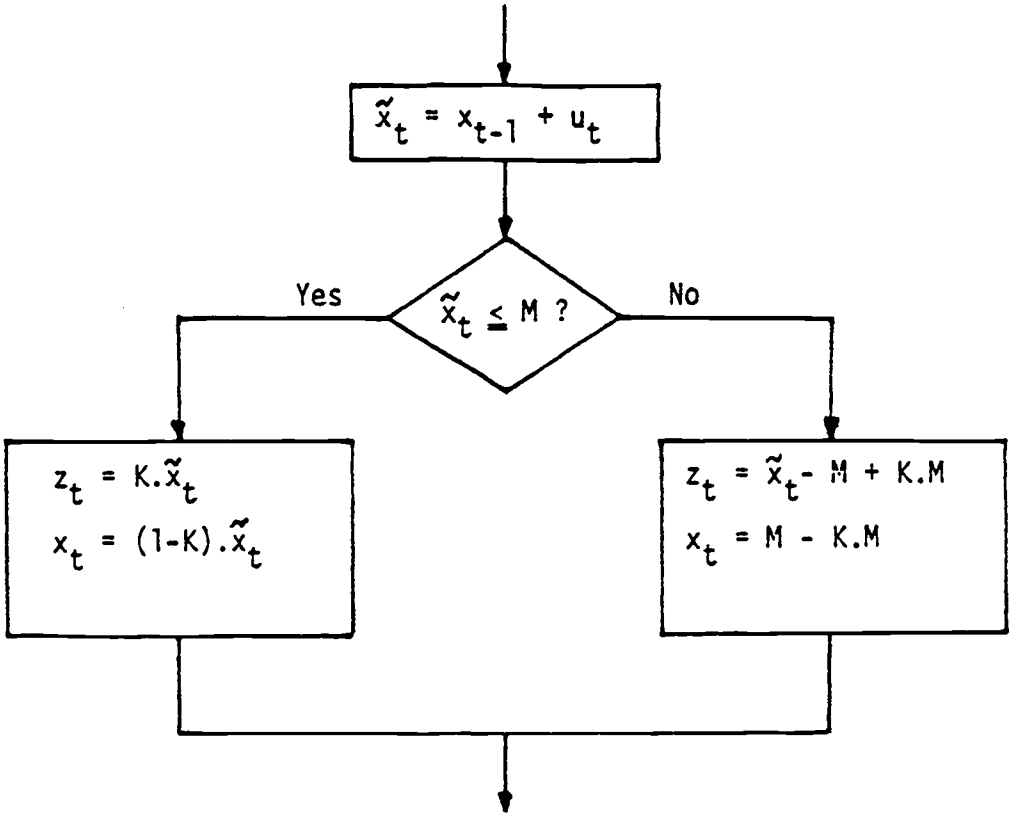


Figure 3.5. Flowchart for the simple single reservoir model

$$T_{1t} = \begin{cases} 1 & \text{for } x_{t-1} \leq (M - u_t) \\ 0 & \text{for } x_{t-1} > (M - u_t) \end{cases} \tag{3.41}$$

$$T_{2t} = \begin{cases} 0 & \text{for } x_{t-1} \leq (M - u_t) \\ 1 & \text{for } x_{t-1} > (M - u_t) \end{cases} \tag{3.42}$$

The equations of the model are therefore:

$$x_t = T_{1t}\,[(1-K)(x_{t-1} + u_t)] + T_{2t}\,[(1-K)M] \tag{3.43}$$

$$z_t = T_{1t}\,[K(x_{t-1} + u_t)] + T_{2t}\,[x_{t-1} + u_t - (1-K)M] \tag{3.44}$$

The derivatives (3.39) and (3.40) become:

$$\frac{\partial x_t}{\partial K} = T_{1t}\,\left[-(x_{t-1} + u_t) + (1-K)\,\frac{\partial x_{t-1}}{\partial K}\right] + T_{2t}\,[-M] \tag{3.45}$$

$$\frac{\partial x_t}{\partial M} = T_{1t}\,\left[(1-K)\,\frac{\partial x_{t-1}}{\partial M}\right] + T_{2t}\,[(1-K)] \tag{3.46}$$

and

$$\frac{\partial z_t}{\partial K} = T_{1t}\,\left[(x_{t-1} + u_t) + K\,\frac{\partial x_{t-1}}{\partial K}\right] + T_{2t}\,\left[\frac{\partial x_{t-1}}{\partial K} + M\right] \tag{3.47}$$

$$\frac{\partial z_t}{\partial M} = T_{1t}\,\left[K \cdot \frac{\partial x_{t-1}}{\partial M}\right] + T_{2t}\,\left[\frac{\partial x_{t-1}}{\partial M} - (1-K)\right] \tag{3.48}$$

The initial conditions to start the recursions in Equations (3.45) through (3.46) are:

x_0 = some initial assumed or known constant value ($0 \leq x_0 \leq M$)

and since x_0 is a constant independent of small changes in the values of the parameters,

$$\frac{\partial x_0}{\partial K} = 0 \tag{3.49}$$

$$\frac{\partial x_0}{\partial M} = 0$$

The behavior of these derivatives over the space of possible parameter values illustrates some well-known properties of the reservoir. Clearly, if the value of M is large enough that the reservoir is never filled, the value of T_{2t} is zero $\forall$ t. Thus, since $\frac{\partial x_0}{\partial M} = 0$, we will have $\frac{\partial x_t}{\partial M} = 0$ and $\frac{\partial z_t}{\partial M} = 0$ for all t. In other words, if the reservoir is never filled, the parameter M can be perturbed by small amounts without affecting the model output and will therefore not be identifiable. This implies that we should always choose a small enough value for our initial estimate of M, such that the reservoir overflows at least once.

Let us assume that the reservoir remains unfilled (in mode one) during time interval $0 < t < nl$, that at time 'nl' the input is large enough to cause the reservoir to overflow, and that the model remains in mode two during time interval $nl < t < pl$. At time pl, the model reverts to mode one. Note that this constitutes a single overflow event. The model equations for this scenario are:

$$x_t = \begin{cases} f(t,K)x_0 + \sum_{\ell=1}^{t} f(\ell,K)\, u_{t+1-\ell} & 0 < t < nl \\ (1-K)M & nl < t < pl \\ f(t-pl+2,\, K)M + \sum_{\ell=1}^{t-pl+1} f(\ell,K)\, u_{t+1-\ell} & pl < t \end{cases} \tag{3.50}$$

and

$$z_t = \begin{cases} K \cdot f(t-1,K)x_0 + K \cdot \sum_{\ell=1}^{t} f(\ell-1,K)\, u_{t+1-\ell} & 0<t<n1 \\ f(t-1,K)x_0 + \sum_{\ell=1}^{t} f(\ell-1,K)\, u_{t+1-\ell} - (1-K)M & t=n1 \\ u_t & n1<t<p1 \\ K \cdot f(t-p1+1,K)M + K \cdot \sum_{\ell=1}^{t-p1+1} f(\ell-1,K)\, u_{t+1-\ell} & p1<t \end{cases} \tag{3.51}$$

where

$$f(t,K) = (1-K)^t \tag{3.52}$$

Differentiating (3.50) through (3.52), the first derivatives become:

$$\frac{\partial x_t}{\partial M} = \begin{cases} 0 & 0<t<n1 \\ (1-K) & n1<t<p1 \\ f(t-p1+2,\ K) & p1<t \end{cases} \tag{3.53}$$

$$\frac{\partial x_t}{\partial K} = \begin{cases} g_1(t,K)x_0 + \sum_{\ell=1}^{t} g_1(\ell,K)\, u_{t+1-\ell} & 0<t<n1 \\ -M & n1<t<p1 \\ g_1(t-p1+2,K)M + \sum_{\ell=1}^{t-p1+1} g_1(\ell,K)\, u_{t+1-\ell} & p1<t \end{cases} \tag{3.54}$$

and

$$\frac{\partial z_t}{\partial M} = \begin{cases} 0 & 0<t<n1 \\ -(1-K) & t=n1 \\ 0 & n1<t<p1 \\ K \cdot f(t-p1+1,\ K) & p1<t \end{cases} \tag{3.55}$$

$$\frac{\partial z_t}{\partial K} = \begin{cases} g_2(t-1,K)x_o + \sum_{\ell=1}^{t} g_2(\ell-1,K)\, u_{t+1-\ell} & 0<t<n1 \\ g_1(t-1,K)x_o + \sum_{\ell=1}^{t} g_1(\ell-1,K)\, u_{t+1-\ell} + M & t=n1 \\ 0 & n2<t<p2 \\ g_2(t-p1+1,K)M + \sum_{\ell=1}^{t-p1+1} g_2(\ell-1,K)\, u_{t+1-\ell} & p1<t \end{cases} \tag{3.56}$$

where

$$g_1(t,K) = \frac{\partial f(t,K)}{\partial K} = -t(1-K)^{t-1} \tag{3.57}$$

$$g_2(t,K) = f(t,K) + K \cdot g_1(t,K) \tag{3.58}$$

The second derivatives of x_t and z_t with respect to the parameters may be derived if necessary.

Consider, in particular, the derivatives (3.55) and (3.56) of the output which are required for the implementation of a derivative-based optimization algorithm. For the threshold parameter 'M', note that $\frac{\partial z_t}{\partial M}$ is non-zero only at the moment of crossover from mode one into mode two (t=n1) and at all times after the switch back into mode one again (t>p1). However, the value of the derivative after the cross-back is not only smaller in absolute magnitude than at time (t=n1), but it also decays away exponentially with successive time steps. The sensitivity of the output z_t to variations in parameter M does not depend on how long the model remains in mode two.

Now consider the scenario in which several overflow incidents (say N in number) occur, the ith overflow beginning at time (t=ni) and ending at time (t=pi) such that (0<n1<p1<n2<p2 ... <ni<pi ... <nN<pN). In this case, the gradient of output $z_t(\underline{\theta})$ to parameter M is given by (note: j=i+1):

$$\frac{\partial z_t}{\partial M} = \begin{cases} 0 & 0<t<n1 \\ -(1-K) & t=ni \\ 0 & ni<t<pi \quad \forall i \\ K \cdot f(t - pi + 1, K) & pi<t<nj \end{cases} \tag{3.59}$$

Assume that the objective function to be used for parameter estimation consists primarily of a sum of squares of errors. Denoting this function by $F(\underline{\theta})$, we have (t^* = number of output data values; t^* = pN say):

$$F(\underline{\theta}) = \sum_{t=1}^{t^*} [z_t(\underline{\theta}) - \tilde{z}_t]^2 \tag{3.60}$$

$$= \sum_{t=1}^{t^*} r_t^2(\underline{\theta}) \tag{3.61}$$

where

$z_t(\underline{\theta})$ = model output at time t;

$\tilde{z}_t$ = measured output at time t; and

$r_t(\underline{\theta})$ = model residual at time t.

Taking derivatives with respect to the parameters $\underline{\theta}$, we get:

$$\frac{\partial F(\underline{\theta})}{\partial \theta_\ell} = 2 \sum_{t=1}^{t^*} r_t(\underline{\theta}) \cdot \frac{\partial z_t(\underline{\theta})}{\partial \theta_\ell} \tag{3.62}$$

$$\frac{\partial^2 F(\underline{\theta})}{\partial\theta_k \partial\theta_\ell} = 2 \sum_{t=1}^{t^*} \left[\frac{\partial z_t(\underline{\theta})}{\partial\theta_k} \frac{\partial z_t(\underline{\theta})}{\partial\theta_\ell} + r_t(\underline{0}) \cdot \frac{\partial^2 z_t(\underline{\theta})}{\partial\theta_k \partial\theta_\ell} \right] \tag{3.63}$$

For $\theta_k = \theta_\ell = M$, we obtain:

$$\frac{\partial F}{\partial M} = 2 \left[-(1-K) \sum_{i=1}^{N} x_{ni-1} + M \cdot \sum_{i=1}^{N} G_i(K,pi,nj) + \sum_{i=1}^{N} H_i(K,pi,nj,\underline{u},\underline{\tilde{z}})\right] \tag{3.64}$$

and

$$\frac{\partial^2 F}{\partial M^2} = 2 \left[-(1-K) \sum_{i=1}^{N} \frac{\partial x_{ni-1}}{\partial M} + \sum_{i=1}^{N} G_i(K,pi,nj)\right] \tag{3.65}$$

where $(j = i+1)$, and

$$G_i(K,pi,nj) = (1-K)^2 + K^2 \sum_{t=pi+1}^{nj-1} f^2(t-pi+1,K) \tag{3.66}$$

$$H_i(K,pi,nj,\underline{u},\underline{\tilde{z}}) = (1-K)\left[u_{ni} - \tilde{z}_{ni}\right] +$$

$$K \sum_{t=pi+1}^{nj-1} \{f(t-pi+1,K) \left[K \sum_{\ell=1}^{t-pi+1} f(\ell-1,K)\, u_{t+1-\ell} - \tilde{z}_t\right]\} \tag{3.67}$$

$\partial F/\partial M$ is in general non-zero whenever the value of M is small enough that the reservoir overflows at least once (as in the derivation). Consider the point $\tilde{M} = \max [x_t(\tilde{M})]$, i.e., the value of M such that the reservoir is just at the point of no overflow. Clearly, if $M > \tilde{M}$, then the derivatives of

$\partial F/\partial M$ and $\frac{\partial^2 F}{\partial M^2}$ (as also $\frac{\partial^2 F}{\partial K \partial M}$) will be zero since the reservoir capacity will never be exceeded. For $M < \tilde{M}$, however, the derivative $\partial F/\partial M$ will be non-zero, except at the optimal value of M, say M*. Hence, in order to find M* using an optimization algorithm, we must ensure that $M < \tilde{M}$ •, or else the algorithm will fail in its attempt to find the best parameter set. This much is fairly obvious. A property of the function $F(\underline{\Theta})$ is that it is approximately quadratic in the threshold parameter M (a similar result was proved by Johnston and Pilgrim (1976) for a nondraining store, i.e., K=0). From (3.65), we see that the second derivative $\frac{\partial^2 F}{\partial M^2}$ consists of a term independent of M and a term directly dependent on M through the derivative of state variable $\frac{\partial x_{ni-1}}{\partial M}$. From Equations (3.52) and (3.53), we have:

$$\frac{\partial x_{ni-1}}{\partial M} = \begin{cases} 0 & i=1 \\ (1-K)^{ni-ph+2} & i>1,\ (h=i-1) \end{cases} \tag{3.68}$$

Note that (ni-ph) denotes the number of time steps between the two successive overflow events. Since (1-K)<1, the value of $\frac{\partial x_{ni-1}}{\partial M}$ is negligible even if the time between overflow incidents is fairly small. Hence, the value $\frac{\partial^2 F}{\partial M^2}$ is approximately constant over values of M (provided $\tilde{M} < M$). It is shown in Appendix 2 that if there are N successive overflow incidents separated in time (i.e., ni-ph is reasonably large), then

$$\frac{\partial^2 F}{\partial M^2} \simeq N \cdot \left[2(1-K)^2 + \frac{2K^2(1-K)^4}{1-(1-K)^2}\right], \quad M < \tilde{M} \tag{3.69}$$

This result indicates that the function $F(\underline{\theta})$ becomes more sensitive to the perturbation of parameter M in a manner roughly proportional to the number of overflow incidents. It is important to emphasize the fact that sensitivity of the output $z_t(\underline{\theta})$ does not depend on how long the model remains in the saturation mode (mode two), but on the number of times the model switches between the two modes. This implies that from the point of view of parameter estimation, a data set containing one large storm event (i.e., $u_t + x_{t-1} > M$) may be less informative than a data set containing many storms of moderate size.

The preceding discussion reveals that when a model contains a threshold element, the optimization problem to be solved will have a continuous objective function that is piecewise continuous in its derivatives. For any optimization algorithm to be useful in solving this problem, we must ensure that the search is not permitted to enter the parameter space for which $\tilde{M} > M$. Model output $z_t(\underline{\theta})$ is essentially a linear function of M; the sensitivity of the function $F(\underline{\theta})$ to M is approximately proportional to the number of times reservoir overflow occurs. $z_t(\underline{\theta})$ and $F(\underline{\theta})$ are complicated nonlinear functions of the parameter K.

3.5.3 A Note on Practical Implementation

In practice, it may be impractical to separate a model into its modes, and to compute the derivatives separately as a function of each mode. While such an exercise may prove fruitful in terms of gaining theoretical insight into the problem, it is not necessary. The derivatives can be obtained simply by differentiation of each subequation as it is encountered in the computer program. The strength of this procedure is that the model need not be modified (computer program rearranged) in order to compute the derivatives.

As an example, in Figure 3.5, we present the methodology as applied to the single reservoir model.

3.6 The Percolation Equation of the NWSRFS Model

3.6.1 Essential Concepts

The representation of the percolation process in the conceptual R-R model is probably the most important part of the model. The percolation process connects the upper and lower zones and helps decide on how much of the water in the upper zone contributes to the quickly reacting components of the storm hydrograph, and how much is transferred to relatively longer-term storage in the lower zone. Thus, it is critical that the exact nature of the nonlinear relationships between the factors contributing to percolation be adequately identified. Strangely enough, as we shall see, the percolation representation is also probably the weakest link in the model structure formulation.

First, let us examine the method by which the percolation volumes for each time period are calculated in a conceptual R-R model. Conceptually, there is considered to be a maximum and a minimum percolation demand. Also, the actual percolation demand at any point in time is related nonlinearly to the "state" of the model at that instant.

Most parametric representations of the percolation process in use today are similar to the well-known Horton Infiltration Equation commonly used to calculate infiltration abstraction from rainfall, so that the overland flow process can be modeled. Horton's Equation is of the form:

$$\text{infiltration demand} \triangleq f_t = f_{min} + (f_{max} - f_{min})\, e^{-kt} \qquad (3.70)$$

where

f_{min} = minimum infiltration demand ($t = \infty$);

f_{max} = maximum infiltration demand ($t = 0$); and

K = rate parameter.

This equation contains three parameters. The equation is usually modified so that it can be used on a digital computer to represent cumulative infiltration demand during a fixed time interval, as this is the way most models are implemented. The modification used is normally similar to one of the following (see Appendix 5):

$$F_n = F_{min} + (F_{max} - F_{min}) \exp \{-X(\frac{UZC_n}{UZM-UZC_n}\} \tag{3.71}$$

or

$$F_n = F_{min} + (F_{max} - F_{min}) \cdot \left(\frac{UZM - UZC_n}{UZM}\right)^X \tag{3.72}$$

where

F_n = cumulative infiltration demand during time interval 'n';

F_{min} = cumulative minimum infiltration demand during a time interval;

F_{max} = cumulative maximum infiltration demand during a time interval;

UZC_n = contents of upper zone at beginning of time interval n;

UZM = maximum storage possible in upper zone; and

X = parameter controlling nonlinearity of the relationship.

The infiltration demand is now dependent on the amount of water present in the zone into which water is infiltrating. So, infiltration demand F_n is maximum when the receiving zone is empty and is minimum when the receiving

zone is full. The quantity F_{min} is not zero because even when the receiving zone is full, water leaving the receiving zone makes way for the same amount of water to enter.

Note that both of the above representations are essentially very similar and that the use of the second form might be considered preferable as no exponentiation is required. It is also important to note that these equations represent infiltration "demand" during a fixed time interval and if this demand is not met due to unavailability of sufficient moisture, then the amount of infiltration actually occurring will be equal to the total moisture available.

As mentioned, the percolation process is usually envisaged to be similar to the infiltration process. It is assumed that maximum and minimum percolation demands exist, and that the amount of percolation demand at any instant depends both on the moisture deficiency of the lower zone and the availability of moisture in the upper zone. There is considered to be a nonlinear relationship between the percolation demand (sometimes called "potential percolation") and the lower zone deficiency. The identification of the extent of this nonlinearity (through estimation of parameter X) is an important part of the parameter estimation problem. As discussed by Bloomfield et al. (1981), a critical assumption made in these equations (3.71 and 3.73) is that the characteristics of the nonlinearity are considered to be independent of the current "state" (storages in different zones) of the model. This assumption is generally made in order to maintain a fairly simple model representation and to keep the dimensionality of the estimation problem low. It can be argued that the more complex representation proposed by Bloomfield et al. (1981) would not help to resolve the parameter observability problem as they theorize. Instead, the higher dimensionality of such

a representation, though conceptually sound, is likely to result in a more difficult estimation problem due to the introduction of relatively insensitive parameters. As was discussed earlier, there may often not be sufficient information content in the data to adequately identify the simpler representation, let alone the more complex ones.

3.6.2 Mathematical Representation of Percolation

The work presented in this report relates directly to the structure of the percolation representation of the Sacramento Model. This is the same formulation as used in the SMA-NWSRFS model (Version II). However, it should be noted that similar representations can be found in other models. "The percolation function (of the Sacramento Model) is considered to be the key element in the transfer of water within the model as it affects water movement throughout the soil profile and is itself dependent on the current state of the storage system" (quote, Brazil and Hudlow, 1981). It is a complex function developed by Burnash, Ferral, and McGuire (1973) relating the capacities and contents of both upper and lower zones and the free water depletion coefficients. The percolation mechanics have been designed to correspond with observed characteristics of the motion of moisture through the soil mantle, including the formation and transmission characteristics of the wetting front, as reported by Hanks et al. (1969) and Green et al. (1970). Hence, it is generally considered to be a "good" representation of the true process. The equation used is given below:

$$\underset{\text{Demand}}{\text{Percolation}} \triangleq \text{PPERC} = \text{PBASE} \cdot [1 + \text{ZPERC}(\text{LZDR})^{\text{REXP}}]\ \text{UZFDR} \qquad (3.73)$$

This represents the percolation demand during a single computational time interval,

where

PBASE = minimum percolation demand when lower zone is full and upper zone is full

$$= (LZFPM \cdot LZPK + LZFSM \cdot LZSK) \tag{3.74}$$

LZDR = lower zone deficiency ratio

$$\frac{(LZTWM + LZFSM + LZFPM) - (LZTWC + LZFSC + LZFPC)}{(LZTWM + LZFSM + LZFPM)} \tag{3.75}$$

UZFDR = upper zone free water contents ratio

$$= \frac{UZFWC}{UZFWM} \tag{3.76}$$

As can be seen, the equation contains a total of eight parameters that must be estimated. They are listed in Table 3.1.

The terms with last letter 'C' constitute the states of storage in the respective reservoirs of the model. They are listed in Table 3.2. The high dimensionality of this equation immediately indicates that strong parameter interaction relationships and compensatory behavior can exist. The equation can be represented as:

$$PPERC = PBASE \left(1 + ZPERC \left(\frac{LZM - LZC}{LZM}\right)^{REXP}\right) \frac{UZC}{UZM} \tag{3.77}$$

where

LZM = LZTWM + LZFSM + LZFPM

UZM = UZTWM + UZFWM

TABLE 3.1

Parameters Present in the Percolation Equation of the SMA-NWSRFS Model

Parameter	Explanation
UZFWM	Upper zone free water maximum (vol. or $\frac{vol}{area}$) - Threshold Parameter
LZTWM	Lower zone tension water maximum (" ") - Threshold Parameter
LZFPM	Lower zone free water primary storage (" ") maximum - Threshold Parameter
LZFSM	Lower zone free water secondary (" ") storage maximum - Threshold Parameter
LZPK	Lower zone primary storage depletion (unitless) constant - Rate Parameter
LZSK	Lower zone secondary storage depletion (") constant - Rate Parameter
ZPERC	A parameter limiting maximum amount (") of percolation
REXP	A parameter controlling the nonlinear (") relationship between the percolation demand and the lower zone deficiency ratio

TABLE 3.2

State Variables Present in the
Percolation Equation of the SMA-NWSRFS Model

State Variable	Explanation
UZFWC	Upper zone free water contents (vol. or $\frac{\text{vol}}{\text{area}}$)
LZTWC	Lower zone tension water contents (" ")
LZFPC	Lower zone free primary water contents (" ")
LZFSC	Lower zone free secondary water contents (" ")

$$LZC = LZTWC + LZFSC + LZFPC$$

$$UZC = UZTWC + UZFWC$$

without losing sight of its basic properties. Here the three lower zone storages have been lumped together as a single element, where LZM is the maximum size of the lower zone storage and LZC is its contents. Similarly, UZM is the size of the upper zone storage and UZC is its contents.

The similarity of this equation to Horton's Infiltration Equation becomes apparent when the equation is written as follows:

$$PPERC = A + B \left(\frac{LZM - LZC}{LZM}\right)^{REXP} \tag{3.78}$$

where

$$PPERC = F_n \tag{3.79a}$$

$$A = PBASE \cdot \frac{UZC}{UZM} = F_{min} \tag{3.79b}$$

$$B = ZPERC \cdot PBASE \cdot \frac{UZC}{UZM} \tag{3.79c}$$

$$= ZPERC \cdot F_{min} = F_{max} - F_{min} \tag{3.79d}$$

and hence

$$ZPERC = \frac{F_{max}}{F_{min}} - 1 \tag{3.79e}$$

$$REXP = X \tag{3.79f}$$

Now, the latter formulation reveals quite clearly that, though the equation of Burnash et al. (1973) is written in a conceptually appealing form, it is

not a form that is desirable from a parameter identification point of view. The parameters ZPERC and REXP are the only two parameters in the equation which are not connected explicitly with the operational mechanics of any other part of the model. REXP essentially determines the degree of nonlinearity in the equation, and ZPERC helps determine the maximum percolation demand F_{max}. From Equation (3.79c), it is easy to see that ZPERC acts only as an arbitrary unitless multiplying constant in the following sense. Rewrite A and B in terms of $\tilde{\tilde{A}}$ and $\tilde{\tilde{B}}$ as follows:

$$A = \tilde{\tilde{A}} \cdot UZC \tag{3.80a}$$

$$\tilde{\tilde{A}} = PBASE \cdot \frac{1}{UZM} \tag{3.80b}$$

$$B = \tilde{\tilde{B}} \cdot UZC \tag{3.81a}$$

$$\tilde{\tilde{B}} = ZPERC \cdot PBASE \frac{1}{UZM} \tag{3.81b}$$

It is clear that, in order to determine the percolation demand at any instant, it is the values of the compounded parameters $\tilde{\tilde{A}}$ and $\tilde{\tilde{B}}$ and the parameter REXP that are important and not the combinations of ZPERC, PBASE, UZM, and REXP that give these values. Given certain values for PBASE and UZM, the value of ZPERC will adjust to the value that gives the best value of $\tilde{\tilde{B}}$. Given different values for PBASE and UZM, a different value of ZPERC will give the same value for $\tilde{\tilde{B}}$. Hence, it is clear that a substantial amount of interaction and compensation is likely to exist among these three parameters of the model. By rewriting the percolation equation in terms of $\tilde{\tilde{A}}$ and $\tilde{\tilde{B}}$, a substantial amount of this interaction should be reduced.

3.6.3 The ZPERC-REXP Relationship

At this stage, the two parameters ZPERC and REXP which affect primarily the fitting of the percolation equation will be examined. A rearrangement of Equation (3.73) gives:

$$ZPERC = \frac{\left[\left(\frac{PPERC}{PBASE \cdot UZFDR}\right)^{-1}\right]}{(LZDR)^{REXP}} \tag{3.82}$$

where

PPERC = percolation demand during a small computational time interval Δt.

The interdependence between the two parameters, ZPERC and REXP, is plotted in Figure 3.6 as the combination of values that will give the same ΔPPERC for a given lower zone deficiency ratio LZDR. As can be observed, the relationship is fairly linear for large values of LZDR. Also, for low values of both parameters, many sets of values could provide reasonably similar results over a broad range of values of LZDR. This raised some suspicion that the superimposed effects of the entire data set might result in substantial parameter interaction. Note that if present, this interaction is expected to be positive. That is to say, the direction of lowest response sensitivity is expected to lie in the first and third quadrants of the two-parameter subspace.

3.6.4 Response Surface Analysis of the ZPERC-REXP Parameter Space Using Real Data

In order to study the ZPERC-REXP relationship further and to check on the suspicions raised by the previous analysis, it was decided to conduct an experimental analysis using the SMA-NWSRFS model.

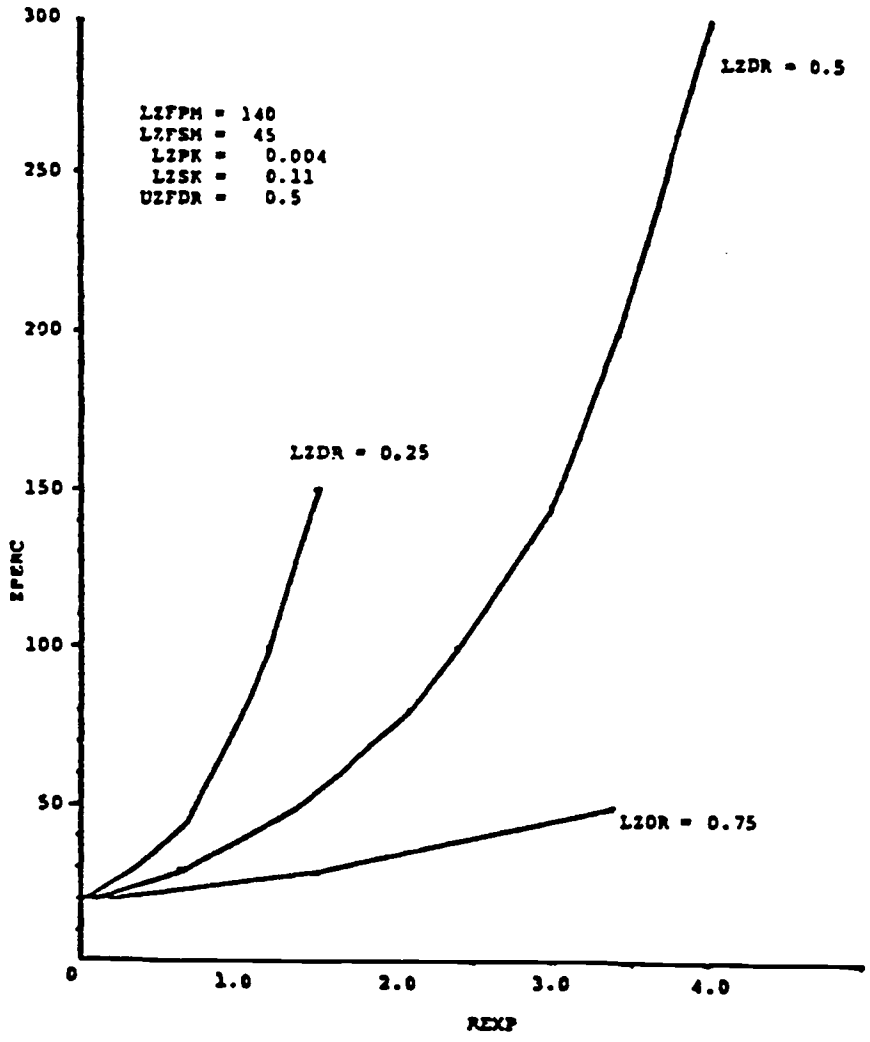


Figure 3.6. Compensating values of ZPERC and REXP for one computational time interval

The ZPERC-REXP parameter space in the region of two-parameter sets obtained by the calibration process was examined. The parameter set was that obtained by Sorooshian et al. (1981) using the SLS[1] criterion and one year (an average year) of calibration data from Leaf River, Mississippi. The parameter values are reported in Table 3.3.

The method for examining the parameter space was as follows. A grid of SLS function values centered on the nominal parameter values was obtained. The step size of each grid axis was chosen equal to one percent of the corresponding nominal parameter value. Then the contour maps were created.

The parameter ZPERC was varied from 190 to 218 in steps of 2.0, and parameter REXP was varied from 0.57 to 0.64 in steps of 0.005. The contour map of the SLS criterion values is presented in Figure 3.7. The point to which the algorithm had converged is marked by a circle. Looking at the SLS criterion contour map, we see that the region is highly irregular. A small elliptical-shaped local optimum is seen in the upper right-hand corner. However, the most important phenomenon that is observed is that the points of minimum function value constitute a long flat valley (dashed line) running right through the region with approximately equal function value at all points along the line. Moreover, this line that denotes the compensating set of parameter values which give the same objective function value is oriented in the expected direction of interaction.

---

(1) SLS - Simple Least Squares Criterion

TABLE 3.3

Parameter Sets in the Region of which the Response Surface Analyses were Conducted

Parameter	SLS calibration parameter values
UZTWM*	0.386
UZFWM	29.190
UZK*	0.973
ZPERC	203.260
REXP	0.609
LZTWM	248.500
LZFSM	13.610
LZFPM	184.900

* Parameters not part of the percolation process equation
UZK = upper zone recession constant (rate parameter)
UZTWM = upper zone tension water maximum (threshold parameter)

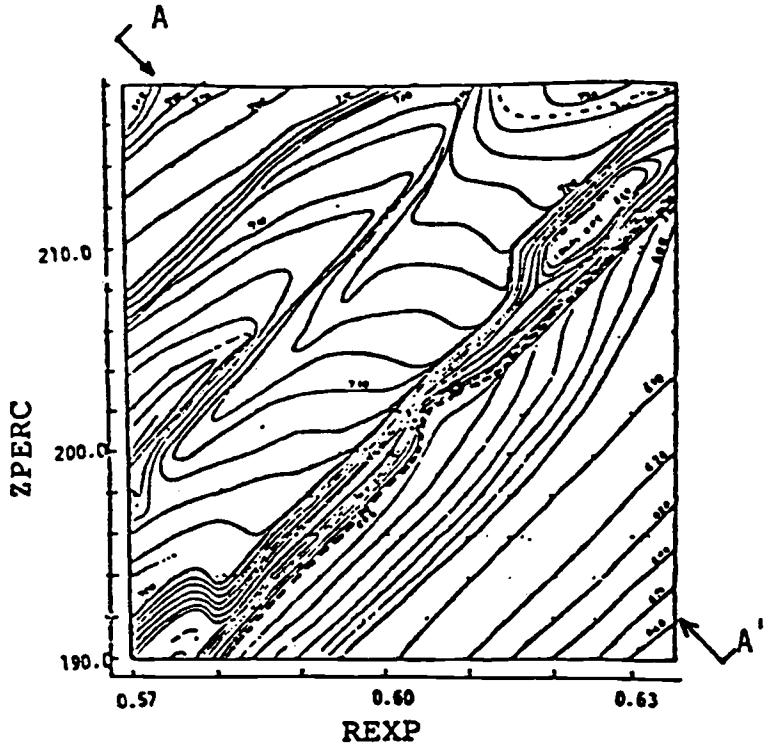


Figure 3.7. SLS criterion contour map in region of SLS criterion "best" parameter set (ZPERC vs. REXP)

3.6.5 The Impact of Percolation "Demand" Computations

As described in the previous sections, the model calculates a percolation demand based on parameter values and existing storages. It then tries to meet that demand from the available upper zone moisture.

If the available moisture is such that the demand is never met, then no change in the values of the parameters deciding that demand will have any effect on the output. On the other hand, if the demand is always met, the output should be quite sensitive to variations in the parameters.

To study the reasons for ZPERC-REXP parameter nonidentifiability, a synthetic study was conducted using the simple six-parameter model SIXPAR (Section 2.3). First, the synthetic rainfall data and the corresponding "true" streamflow data were input to the model and the graphs of both percolation demand and actual percolation were obtained for Z = 50.0 and X = 3.0 (see Figure 3.8). It was found that in about 34% of the computational time intervals, the true value of the percolation is less than the percolation demand. During these time intervals, the process is insensitive to small changes in the parameter values. The critical parameters here are Z and X. The parameter Z essentially determines the maximum percolation demand, and X mainly determines the nonlinearity present in the process. Since in the above case the percolation demand exceeds the actual percolation during a substantial amount of the time, it seems reasonable to expect that the output is relatively insensitive to the value of parameter Z.

A Z versus X contour map of the SLS criterion in the region of the optimum was obtained by perturbing the parameters (Figure 3.9). The The expected elongated valley is present and is oriented close to the direction of the X axis indicating lower sensitivity of the response to Z.

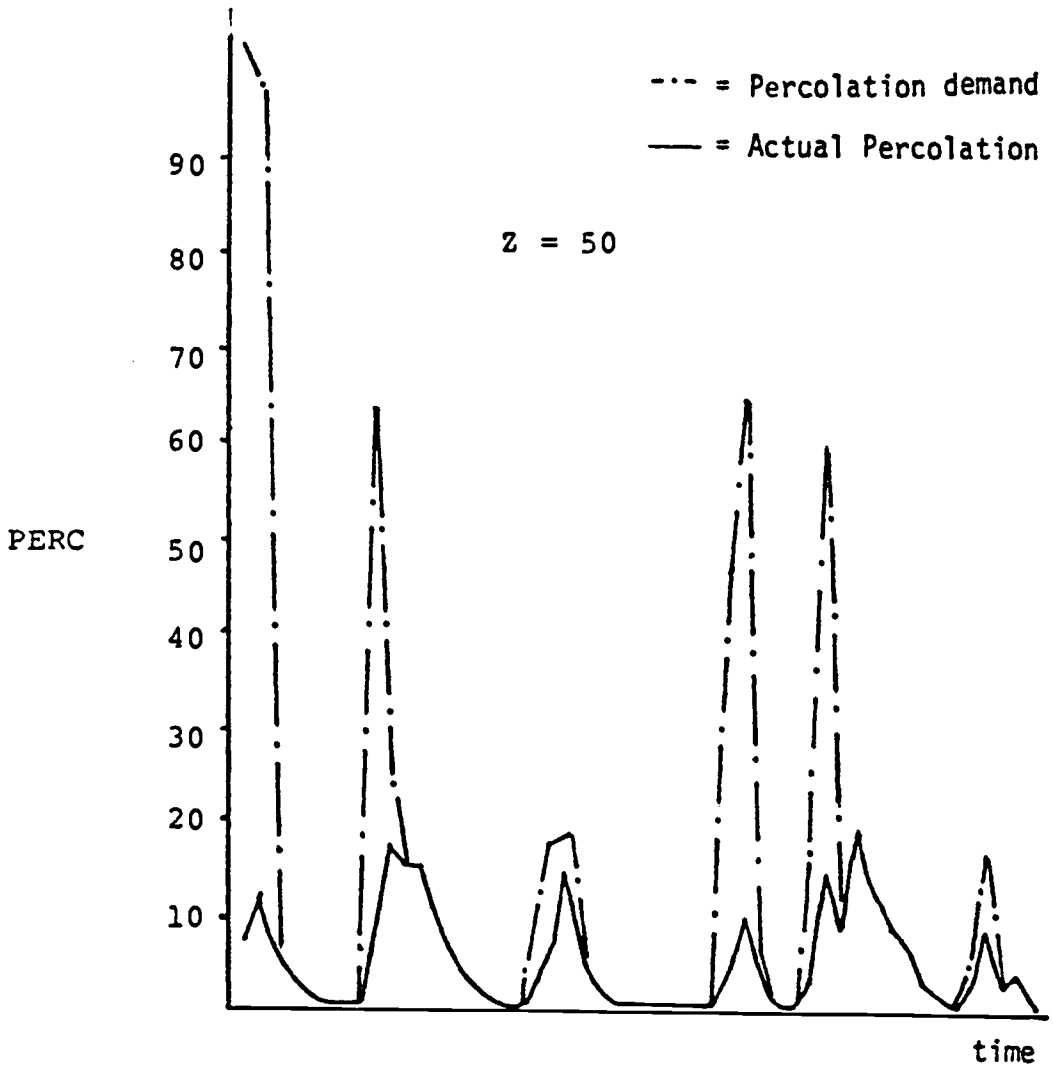


Figure 3.8. Percolation demand and actual percolation values for the six-parameter model for Z = 50

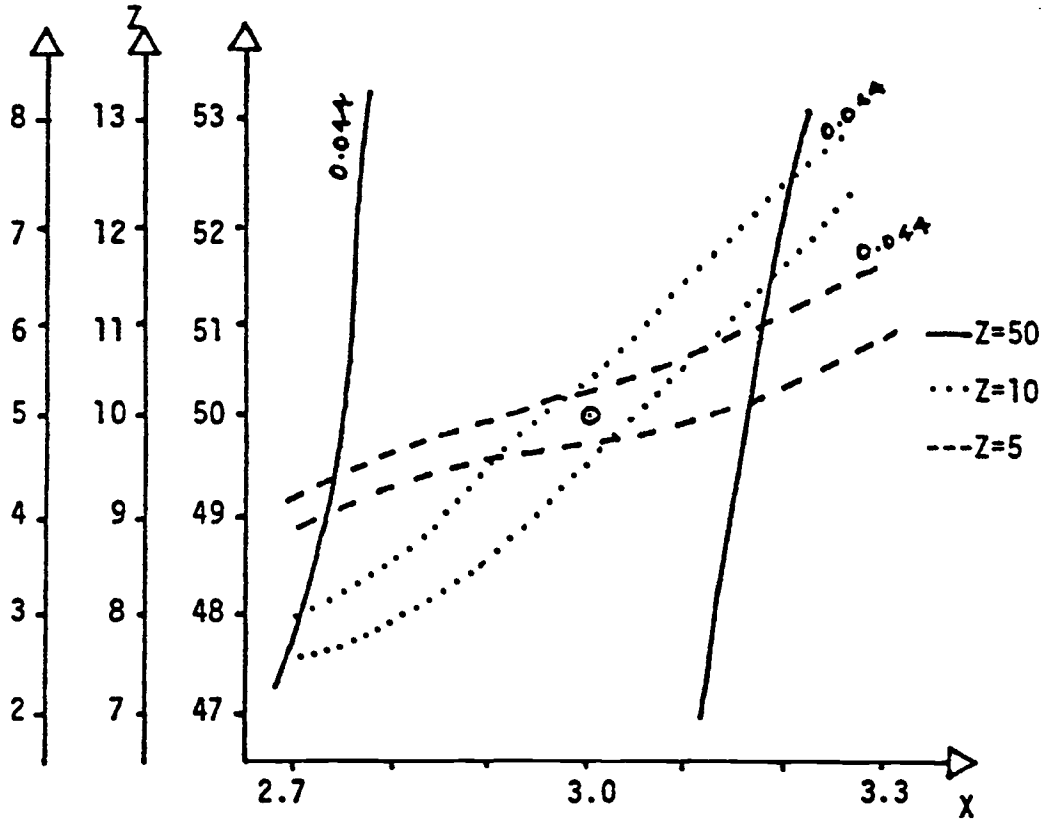


Figure 3.9. SLS contour plots of value 0.044 for the three cases of Z = 50, 10, and 5, showing nature and orientation of the valley (Z versus X)

Now, if the percolation demand is always met, then the sensitivity to Z should be increased and the extended valley eliminated. To test this, the value of Z was lowered to reduce the percolation demand values. Values of Z = 10.0 and 5.0 were tested. The results appear in Figures 3.10a,b and 3.9. At Z = 10.0, the percent number of computational time intervals in which the demand exceeds availability drops to 18% (Figure 3.10a) and the contour plot (Figure 3.9) indicates higher sensitivity to both Z and X. When Z = 5.0, the percent number of intervals in which that demand exceeds availability is only 10% (Figure 3.10b), and the response (Figure 3.9) is relatively more sensitive to Z. In both cases, however, the extended interacting valley is still present but with a different orientation in each case.

Finally, high resolution contour plots were obtained for the Z = 50.0 and Z = 5.0 cases (Figure 3.11a,b). In both cases, no indication of closed contours was detected. Hence, it would seem that the extended interaction valley is inherent to the model structure, and that its orientation depends on the degree of "parameter activation" that is brought about by the data used.

3.6.6 Implications of the Results to Parameter Observability

The results presented in the preceding section raise some questions concerning the ability to obtain a unique and consistent parameter set using historical data no matter which estimation technique is used--be it manual calibration or automatic parameter estimation. In either case, an initial set of parameters must be specified. If the automatic method is used to estimate Z and X, the search technique will terminate at the first point at the bottom of the long and narrow valley which it encounters. It should be borne in mind that, due to errors in the data and/or the inability of the

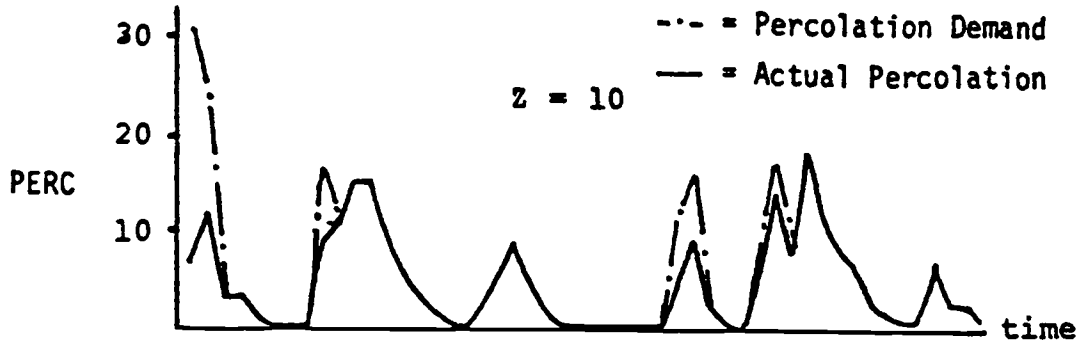


Figure 3.10a. Percolation demand and actual percolation values for the six-parameter model for Z = 10

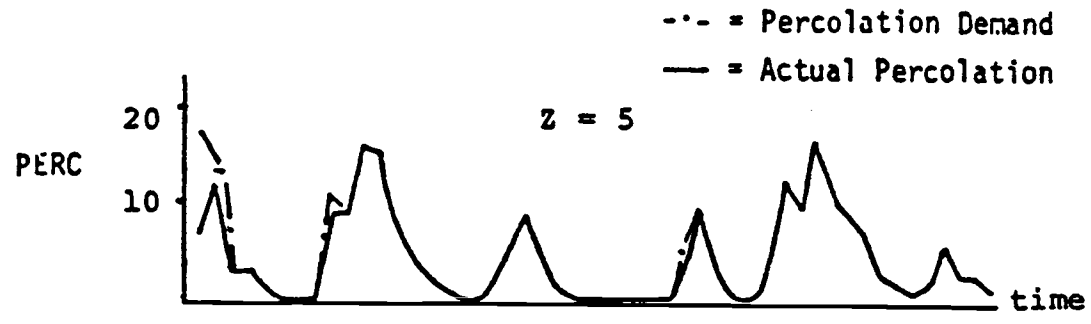
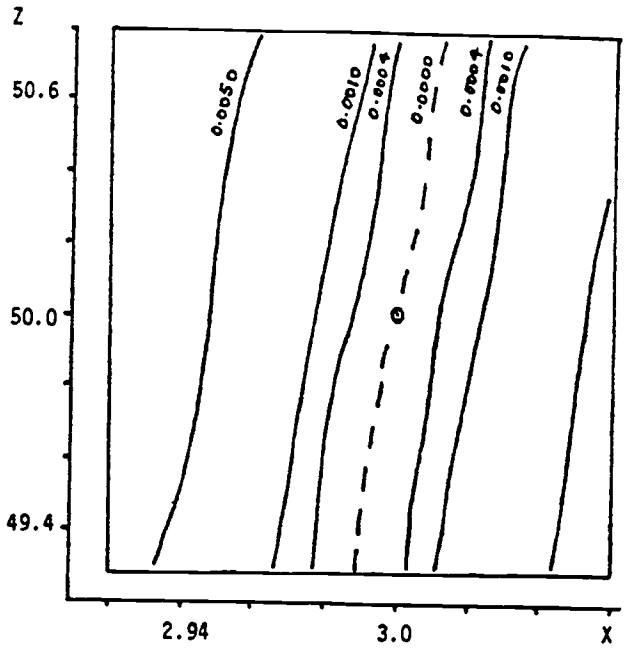
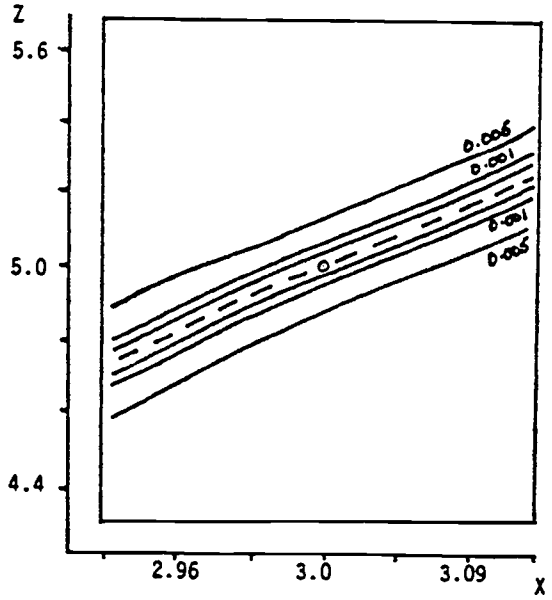


Figure 3.10b. Percolation demand actual percolation values for the six-parameter model for Z = 5



Criterion = SLS

Figure 3.11a. Higher resolution contour plot for case when Z = 50 showing nature and orientation of ridge (Z versus X)



Criterion = SLS

Figure 3.11b. Higher resolution contour plot for case when Z = 5 showing nature and orientation of ridge (Z versus X)

model to properly simulate the true process, the criterion value may vary from point to point so that what we have is an elongated valley-like phenomenon. Due to the inability of the search technique to proceed, termination of the optimization will occur prematurely and at different parameter sets for different starting values.

At this point, the question may be raised as to why this is of any importance at all and, that if any set of values of Z and X on the floor of the elongated valley is as good as any other (from a mathematical point of view), then why the concern? The answer is that different sets of values of the parameters are not as good as each other due to two very basic reasons.

To understand the first reason, let us pretend we know the true parameter set. Due to the nature of the data being used, there will exist an elongated valley on the Z-X response surface oriented in some direction, this direction being dependent on the data. Let us assume that we select a point on the floor of the valley at some distance from the true optimum (see Figure 3.12). Now a second data set is selected to be used as a verification data set. If the characteristics of this data are different (wetter or drier, for example), the valley will still pass through the true parameter set but will be oriented differently. Hence, the selected parameter set will no longer be part of the indifference set for the new data and, if used, will give nonoptimal forecasting results, due to poor reproduction of the percolation process.

The second reason concerns a topic only briefly mentioned earlier, i.e., the high degree of parameter interdependence that exists as a result of the nature of the mathematical formulations. Parameter interdependence implies that, if there is an error in the estimation of one parameter, there will be errors in the estimated values of the other interacting parameters

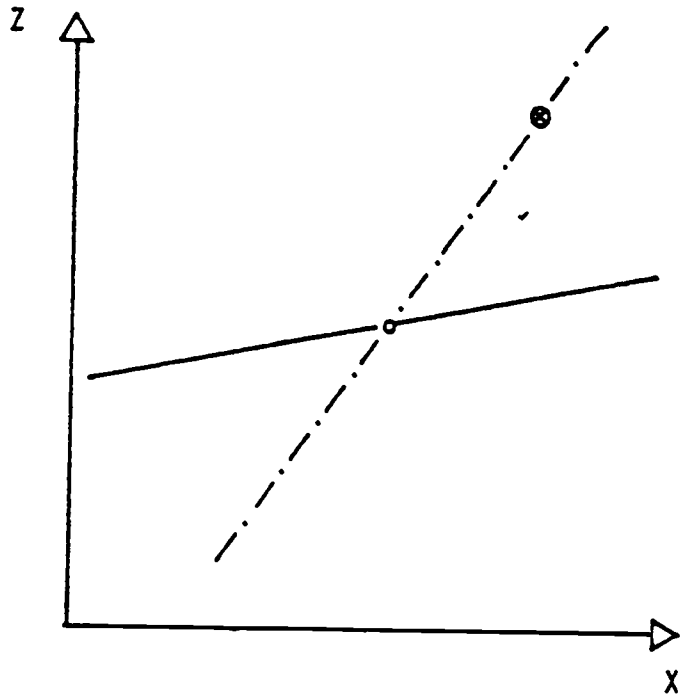


Figure 3.12. Illustration of effect of choosing incorrect parameter set from calibration data valley

— valley for calibration data set

-•- valley for forecast data set

⊙ "true" parameter set

⊗ point of termination of search procedure during calibration

as a result of the compensating changes required to minimize the total error. Of course, the magnitude of the errors caused in the values of the other parameter depends on the extent and importance of the interaction. A look at the SMA-NWSRFS percolation function reveals the presence of eight parameters interacting directly through this process. As shown earlier, the identification of the "true" values of the ZPERC and REXP parameters is a difficult, if not impossible, task. If at the same time, the two parameters have significant interactions with the other model parameters (as they are expected to have), errors in the estimation of ZPERC and REXP will cause errors in the estimation of the other interacting parameters. Whether a manual or an automatic technique is used, parameters identified are likely to be unsatisfactory and certainly nonoptimal.

As an example of the implications of these facts to the efficiency of automatic search techniques, a limited study was carried out on the SMA-NWSRFS model. The same data set and "optimal" parameter values described in Section 3.6.4 for the SLS case were employed. Figure 3.13 is a plot of the LZFPM versus LZFSM contour map of the SLS criterion in the region of the SLS optimal set of parameters. Figure 3.14 is a plot of the same region when ZPERC is changed from 200 to 194, and REXP is changed from 0.61 to 0.59. The plot indicates very clearly that, when we move to the new point along the line of equal function values (see Figure 3.7), the contours get much wider, and lower function values are obtained towards the right-hand side (compare Figure 3.14 to Figure 3.13). Thus, due to poor estimation of ZPERC and REXP caused by premature termination of the search technique at the bottom of the extended valley, both LZFSM and LZFPM have been poorly estimated.

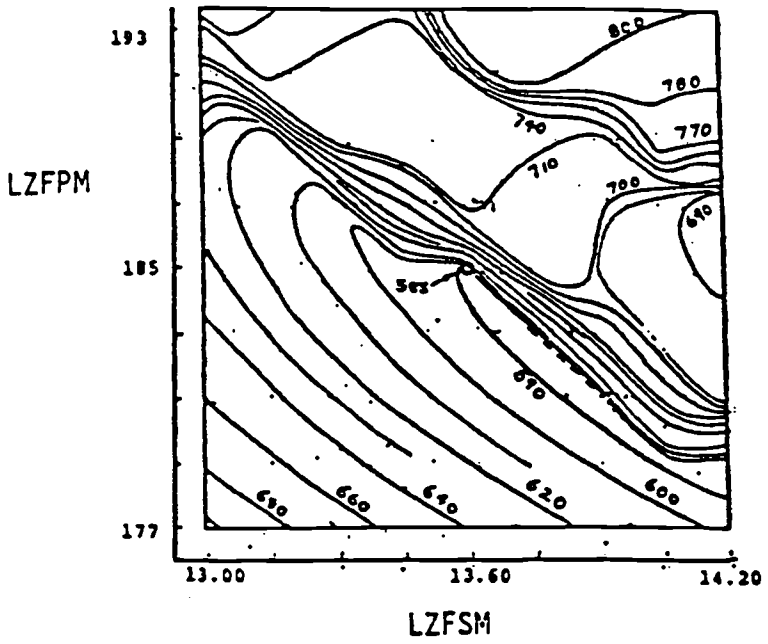


Figure 3.13. SLS contour map for ZPERC = 203, REXP = 0.61

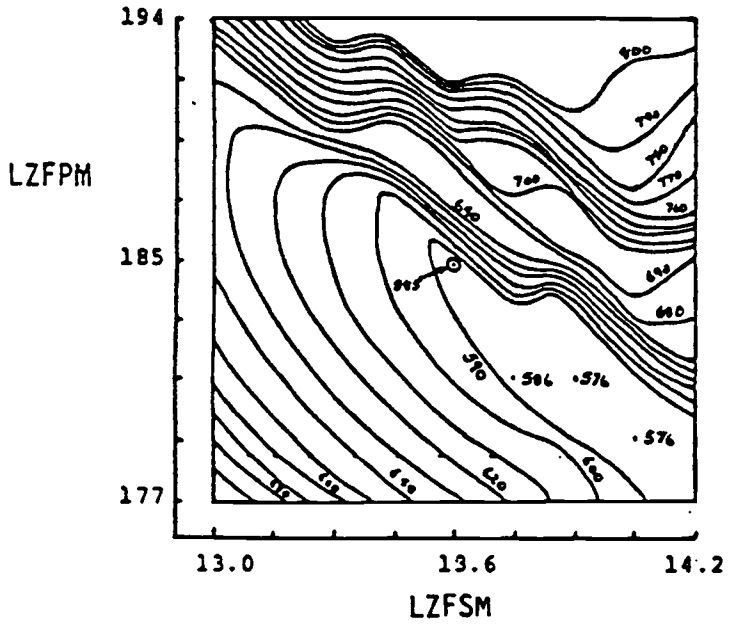


Figure 3.14. SLS contour map for ZPERC = 194, REXP = 0.59.
(LZFPM versus LZFSM)

3.7 Reparameterization of the Percolation Equation

3.7.1 Introduction

In Sections 3.1 through 3.3, the concept of structural identifiability was discussed. In Section 3.6, we have detected that the NWSRFS model is structurally nonidentifiable or near-nonidentifiable. In order to have any confidence in the identified model, this problem must be resolved. Now, the index of structural identifiability is the second derivative matrix given in Equation (3.7). Note here that since $z_t(\underline{\theta})$ is actually a function of both $\underline{\theta}$ and the chosen input data sequence $\underline{u}(t) = \{u_1, \ldots u_t\}^T$, so too is the structural identifiability matrix $\nabla^2_{\theta} SI(\underline{\theta})$. Hence, one alternative is to attempt to use a different input data sequence and see if that improves matters. In cases where one is free to choose the input data sequence, this method could prove successful. However, in watershed models, such freedom is usually quite limited. Moreover, it is possible that no improvement may be observed over the range of feasible data sets. Another alternative is to restrict the model to one that is structurally identifiable using the available input data. Note that the simplest way to do this is to assume that some of the parameters are known and to fix them at some nominal values. However, to do this, one must have some suitably reliable qualitative information regarding the values at which they should be set. Further, the solution of the restricted problem will most likely be suboptimal in relation to the actual maximum likelihood values of the parameters. A more suitable solution that may, in some cases, result in improved structural identifiability of the model set is reparameterization of the model.

3.7.2 Theoretical Basis for Reparameterization

Reparameterization involves the selection of a transformation in the parameter space, i.e., the selection of a new parameter vector $\underline{\phi} = R(\underline{\theta})$ such that the model set $M(\underline{\phi})$ has improved parameter identifiability over the model set $M(\underline{\theta})$. It is important to note that reparameterization does not imply that the model structure or behavior need to be changed or simplified, although model simplification or restrictions to the feasible model set can be viewed as subsets of the set of available reparameterization alternatives. The mathematics involved in reparameterization are described below.

Consider a model $M: \Theta, U \rightarrow Z$, parameterized by a set $\theta \in \Theta$. Let $R: \Theta \rightarrow \Phi$ be a mapping from the space of parameters $\underline{\theta} \in \mathbb{R}^p$ to the space of a new parameter set $\underline{\phi} \in \mathbb{R}^p$, where $\mathbb{R}$ is the reparameterization operation and hence $\underline{\phi} = R(\underline{\theta})$. The reparameterized model is then given by $M: \Phi, U \rightarrow Z$. Using the chain rule for differentiation, it is easy to show (note that $\nabla_\theta SI$ = zero) that:

$$\nabla_\phi^2 SI = \nabla_\phi \theta^T \cdot \nabla_\theta^2 SI \cdot \nabla_\phi \theta$$

It is desirable that the chosen reparameterization operation R be chosen to be one-one invertible since, in this case, the solution in the $\underline{\phi}$ space can be translated into the solution in the $\underline{\theta}$ space (possibly for interpretation of results). It is interesting to note that if $\hat{\underline{\phi}}$ is the maximum likelihood value of $\underline{\phi}$, then by the operation $\hat{\underline{\theta}} = R^{-1}(\hat{\underline{\theta}})$, we can obtain $\hat{\underline{\theta}}$, the maximum likelihood value of $\underline{\theta}$.

The objective of the reparameterization operation is to improve the structural identifiability of the model set. Hence, we shall restrict ourselves to "allowable" reparameterizations, i.e., those which "improve" the

properties of the structural identifiability matrix. In particular, we shall be interested in reparameterizations for which,

$$n_i(\phi) < n_i(\theta) \qquad \forall_i \tag{3.83}$$

This is the sensitivity ratio discussed in Section 3.3.3 and will be used as the index to measure the improvement in overall identifiability of the model.

3.7.3 Methods of Applying Reparameterization

There are two methods by which the chosen model reparameterization can be implemented. They are:

(1) Rewriting of the model computer program in terms of the newly-parameterized structural equations.

(2) Retaining the model equations in terms of the original parameterization and rewriting only the derivatives of the model output with respect to the new parameters by means of the chain rule of calculus, i.e.,

Step 1:
$$\underline{\theta} = R^{-1}(\underline{\phi}) \tag{3.84}$$
$$z(\underline{\theta}) = M(\underline{\theta},U)$$

Step 2:
$$\nabla_{\phi}\underline{\theta} = \frac{d}{d\phi} F^{-1}(\underline{\phi}) \tag{3.85}$$

$$\nabla_{\theta} z(\underline{\theta}) = \frac{d}{d\underline{\theta}W} M(\underline{\theta},\underline{u})$$

Step 3:
$$\nabla_{\phi}\underline{z}(\underline{\phi}) = \nabla_{\theta}\underline{z}(\underline{\theta})\ \nabla_{\phi}\underline{\theta} \tag{3.86}$$

Note that all that is required is the inverse of the reparameterization function and its first derivatives. The vector $\nabla_{\phi}\underline{z}$ is then used to construct the structural identifiability matrix using:

$$\nabla^2_{\phi} SI(\underline{\phi}) = \nabla_{\phi}\underline{z}(\underline{\phi}) \cdot \nabla^T_{\phi}\underline{z}(\underline{\phi}) \tag{3.87}$$

Of the two methods, the first is reasonable only for small methods which can be easily programmed. Otherwise, it entails an inordinate amount of extra programming work. The second method, on the other hand, is extremely simple to implement. Further, using method (2), alternative parameterizations can be tested with very little effort.

3.7.4 A Theoretical Look at the Percolation Equation

In Section 3.6, the form of the percolation equation was discussed. In particular, it was discussed that by appropriate grouping of the parameters which were multiplied and added together and by compounding them to form new parameters, one could substantially reduce the degree of parameter interaction present in the model. However, the discussions in Section 3.6 indicate that the major problem appears to come from interaction between the two parameters ZPERC and REXP. Since these parameters are neither multiplied nor added together, the aforementioned approach cannot be used to reduce the interaction between them. Therefore, a reparameterization is sought which is consistent with the mechanics of the percolation process equation. For this analysis, the model SIXPAR will be used.

To illustrate the problem, a graphical representation of the percolation demand and real percolation versus the change in lower zone deficiency ratio (LZDR) is shown in Figure 3.15 for the cases of UC = UM and UC = M/2.

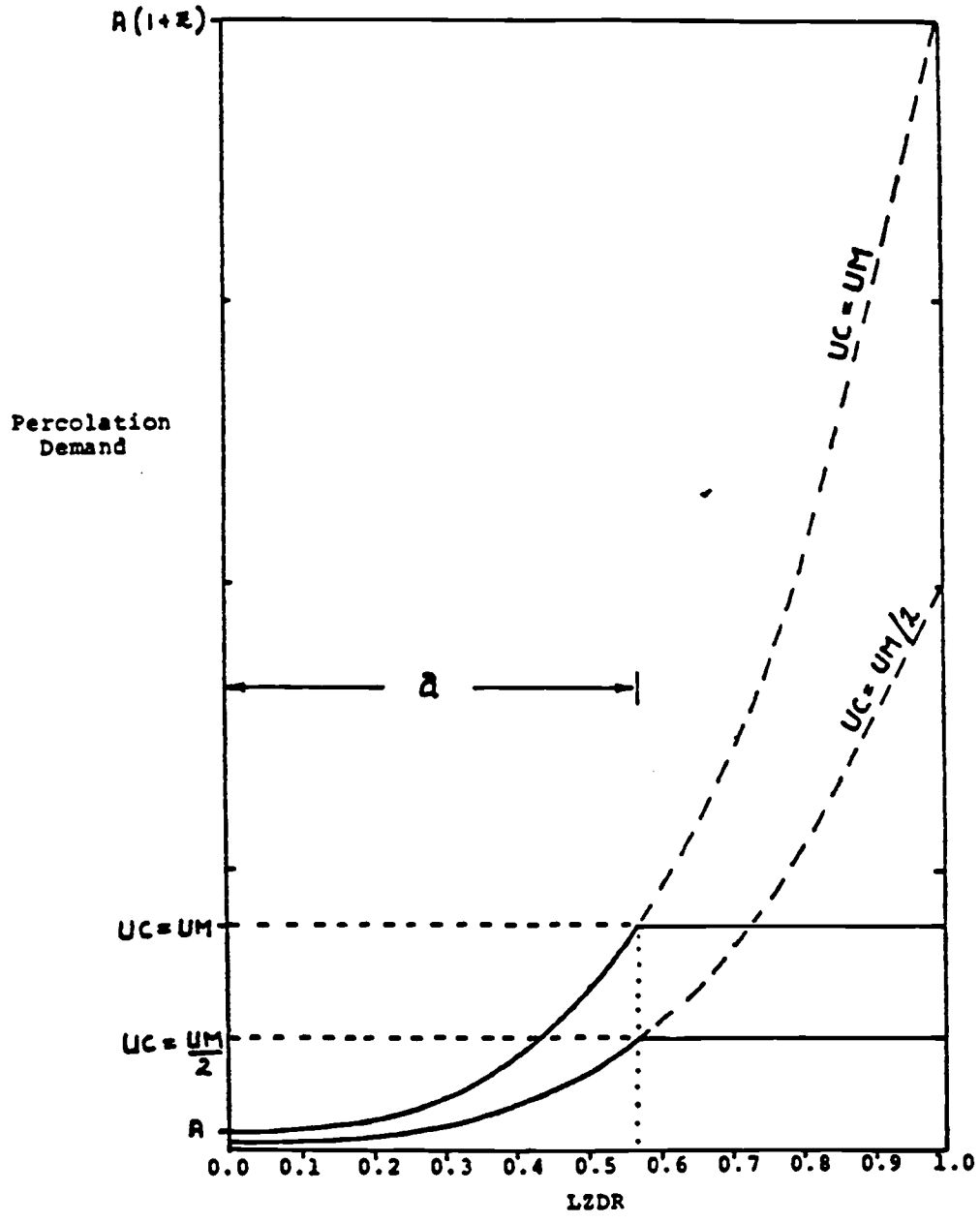


Figure 3.15. Representation of percolation demand as a function of lower zone deficiency ratio

The curved lines represent the percolation demand for each of the cases, while the horizontal line represents the amount of water available in the upper zone. The maximum percolation that can occur is equal to UM, which is the maximum capacity of the upper zone. Hence, percolation demand will always be met for LZDR less than or equal to some value "a" ($0 < a < 1$) and will never be met for LZDR greater than "a". (Note: An important assumption here is that the maximum percolation demand exceeds the size of the upper zone storage UM. If not, the percolation demand will always be satisfied by available upper zone water, and a parameter observability problem would probably not exist).

Thus, irrespective of the data selected for the estimation procedure, the parameters of the percolation equation (Z and X, or ZPERC and REXP) are only influenced by those time periods for which LZDR is less than or equal to "a" ($LZDR < a$). This means that the amount of data directly influencing the estimation of Z and X (ZPERC and REXP) is much reduced. Note also that if UM is small, then so is "a", and the amount of "activating" data will be even less.

We now examine how this affects the observability of Z and X (the results follow in a parallel manner for ZPERC and REXP), and in particular why this should give rise to the elongated interacting valley. From Figure 3.16, it is easy to see that fairly large changes in Z do not cause large deviations of the percolation curve for $LZDR < a$. These deviations can be minimized by an appropriate compensating change in X, and the resulting errors in computed percolation can become quite small. These errors then pass through the lower zone storage reservoirs before they reach the output. The damping effects of the reservoirs ensure that the effects of the errors are very small and hence do not appreciably affect the objective function.

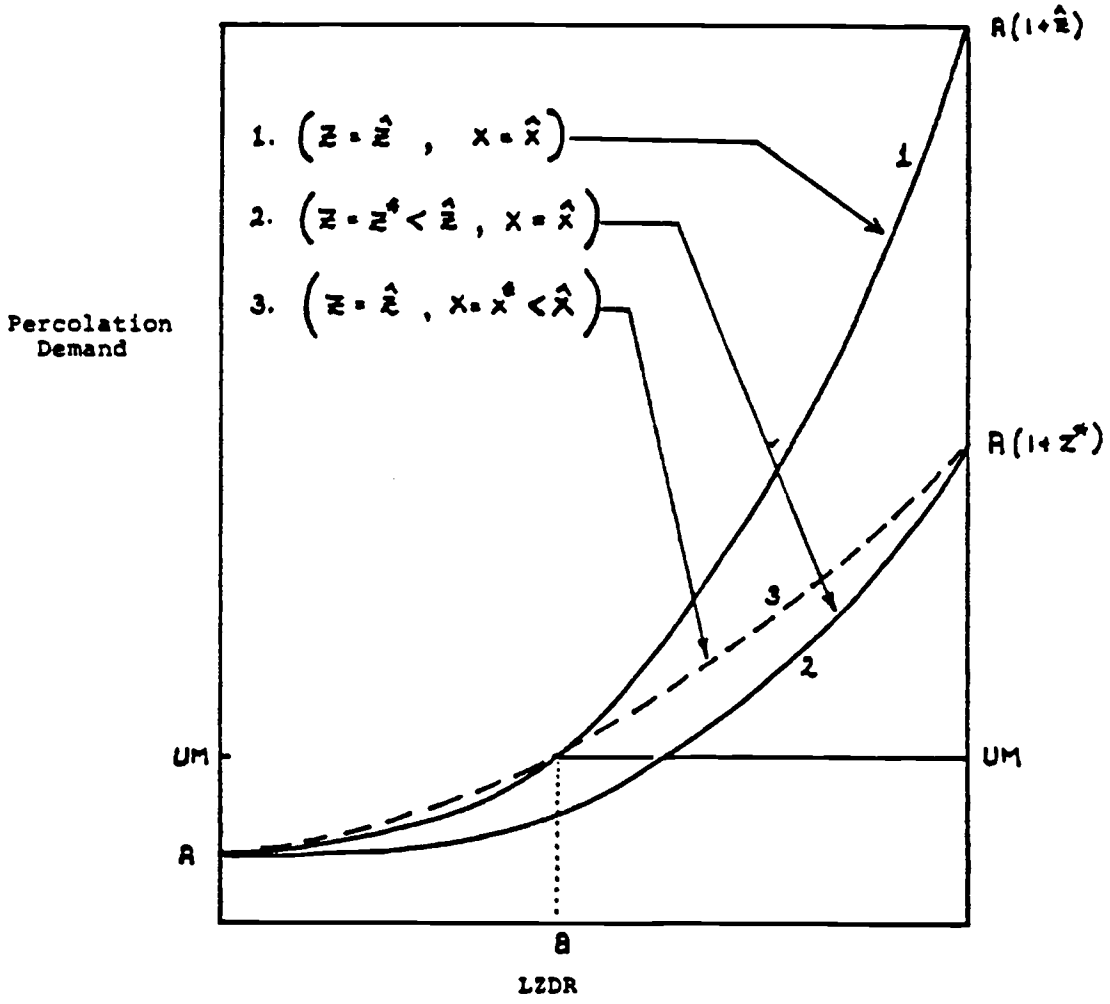


Figure 3.16. Example showing the effect of Z versus X parameter interaction on the percolation demand curve

Z, X̂ = original values

Z*, X* = new values

As a result, the response surface will have an interacting valley in the Z-X plane with virtually equal function values, formed by an extremely elongated contour "ellipse" with its ends well outside the region being examined.

It is obvious that, in order to make the parameters more observable, an appropriate reparameterization is required so as to make the response surface better conditioned. In particular, a parameter is required which is a function of both Z and X and which represents the direction along the valley floor. This parameter should replace the relatively insensitive parameter Z. Proper scaling of the new parameter should allow reasonably elliptical contours of the objective function in the neighborhood of the "best" parameter set. Another factor that must be borne in mind is that the parameter chosen should preferably have some conceptual significance.

From Figure 3.15, it is clear that there is one invariant present which is unaffected by the amount of water available for percolation. This is the value "a". Clearly, "a" represents the value of LZDR at which the percolation demand and available water curves intersect. This value has some nice properties:

(1) It is scaled between zero and one. This means that a reasonable initial estimate for "a" can be made because it is bounded. Z, on the other hand, can vary between zero and infinity.

(2) It can be easily used to calculate approximately how much information was in the data which was useful for estimating the percolation process parameters.

The scaled property means that large changes in Z correspond to much smaller changes in "a" (see Figure 3.16, thus making "a" far more sensitive

than Z. It is clear from Figures 3.15 and 3.16 that an extremely large amount of data would be required to make Z better determined because of the fact that UM is in most cases going to be less than A(1+Z), i.e., maximum percolation demand. Even if the conditions of upper zone being full and lower zone being empty are ever experienced (which is very unlikely), the actual percolation taking place is controlled by the parameter UM and not by the parameter Z.

The relationship between the parameters "a", Z, and X, is now derived. The percolation equation is:

$$Perc = A.[1 + Z.(LZDR)^{X}].\frac{UC}{UM} \tag{3.88a}$$

where A = (LM . LK).

Let the contents of the upper zone be equal to UC. Percolation demand will be equal to UC when LZDR = "a". Hence,

$$UC = A.[1 + Z.(a)^{X}].\frac{UC}{UM} \tag{3.88b}$$

Simplifying:

$$a = [\frac{UM - A}{A.Z}]^{1/X} \tag{3.89}$$

or:

$$Z = [\frac{UM - A}{A.a^{X}}] \tag{3.90}$$

Substituting for Z in the percolation equation (3.88):

$$\text{Perc} = \begin{cases} \left[A + \left(\frac{UM - A}{a^X}\right).\ (LZDR)^X\right].\frac{UC}{UM} & \text{for all LZDR} < a \\ UC \qquad \text{for all LZDR} > a. \end{cases} \tag{3.91}$$

Notice that, due to this reparameterization, there is no need to calculate a percolation demand value, as the true value of percolation depends simply on whether the LZDR at any time is greater than or less than parameter "a".

3.7.5 Response Surface Analysis

The reparameterized percolation equation was programmed into the model SIXPAR in place of the original equation. Using the same parameters as in the earlier synthetic study (Z = 50), the calculated value of "a" is 0.31. The input-output data were used to generate the "a"-X contour map of the SLS criterion, based on equal percent changes of the nominal values of "a" and X (see Figure 3.17). The response is seen to be very sensitive to changes in "a" but relatively insensitive to changes in X. A similar contour map for "a" = 0.6 is shown in Figure 3.18.

At first glance, these results appear to be rather disappointing, as in both cases an elongated direction of low sensitivity is present. However, an encouraging observation is that in both cases, the direction of low sensitivity exhibits very low interaction. We observe that the response is very sensitive to percent changes in the value of "a", but is relatively insensitive to percent changes in the value of X. Obviously, a rescaling of either of these parameters is required so that more circular contours are

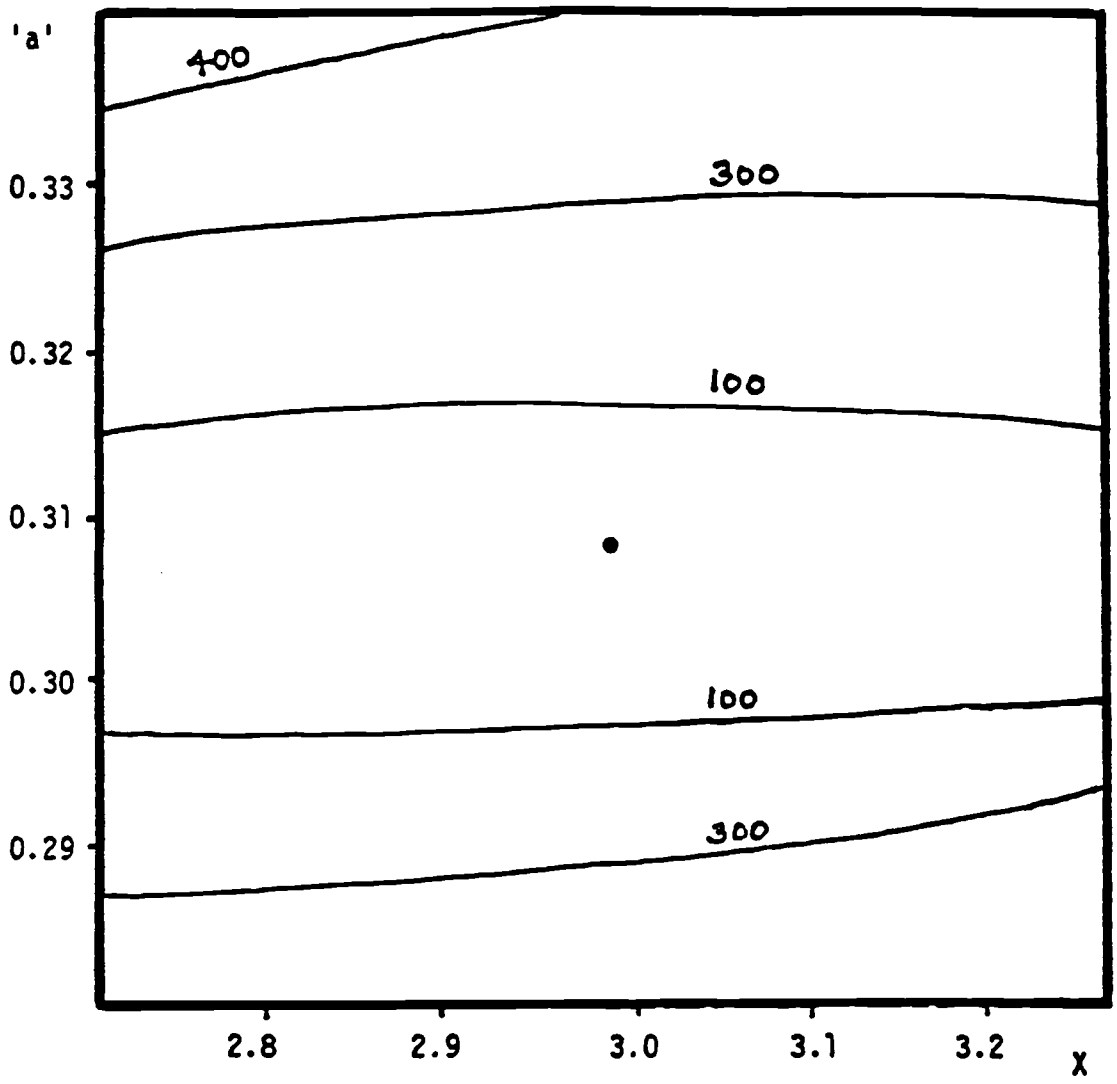


Figure 3.17. SLS contour map of a - versus - x for nominal a = 0.31 and x = 3.0

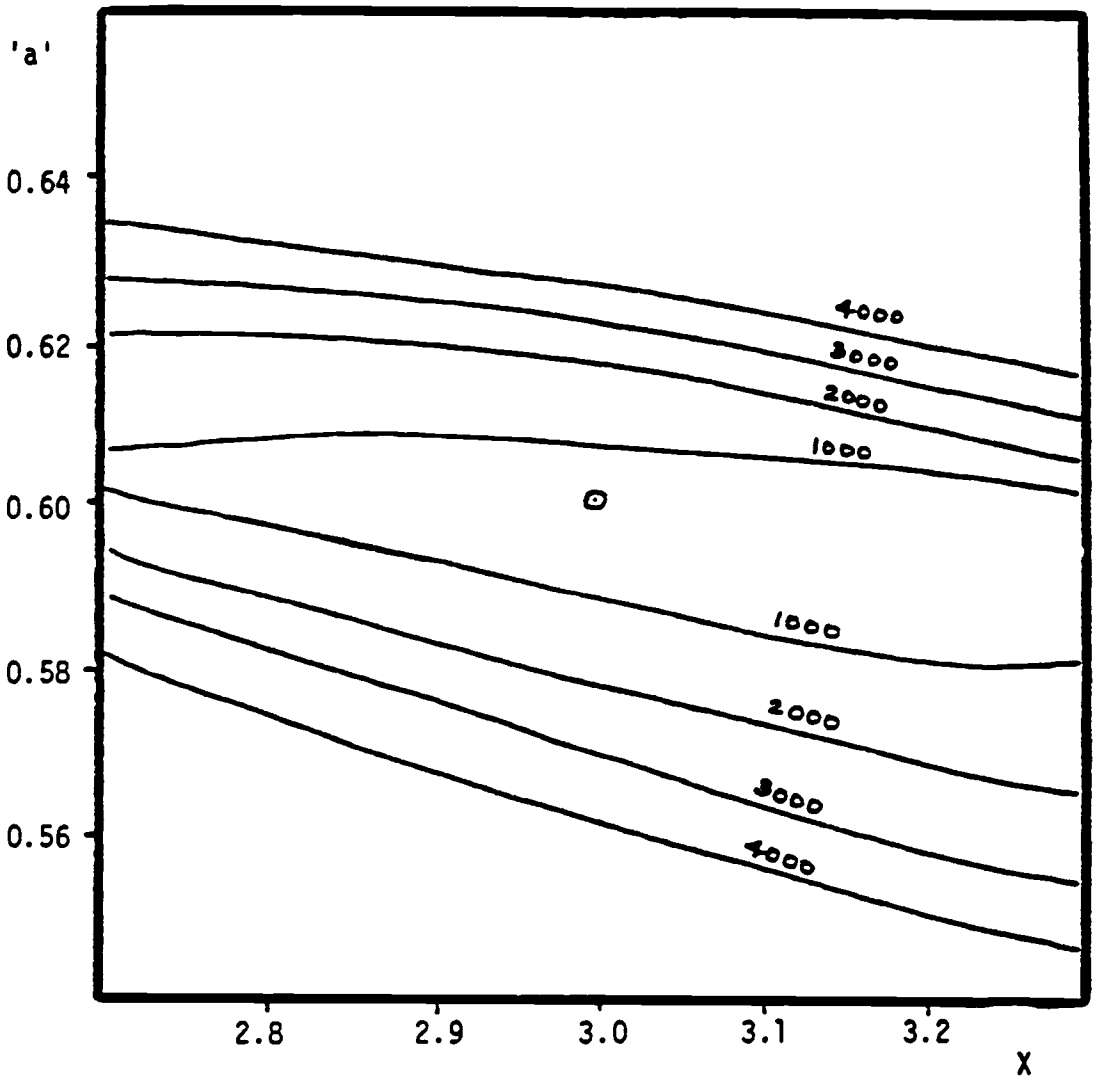


Figure 3.18. SLS contour map of a - versus - x for nominal a = 0.6 and x = 3.0

obtained. This should not be difficult in this case, as the angle of interaction is small.

The important result of this reparameterization which should be emphasized here is that it has reduced the parameter interaction to a minimal value, thus enabling a proper scaling of the parameters to be employed to change the contour shapes. In the case of the Z-X parameterization, the degree of interaction was significant and varied substantially. Hence, in that case, scaling to make the contours more circular was not possible (i.e., when interaction is large, rescaling of a parameter only changes the angle of interaction and does not help to reduce the length of the valley significantly).

3.7.6 Rescaling of Parameter "a"

If the search technique uses a constant step size, there is no problem regarding the selection of an appropriate scale for "a", as a relatively smaller step size can be chosen. However, it is common, as in the application of the SMA-NWSRFS model, that step sizes which are percent changes of the existing parameter values are used. Thus, step sizes increase as the parameter size increases, and vice versa. In this case, a reparameterization "R" is required for "a" such that close to the "best" set, the response of the objective function to a percent change in the value of "R" is similar to the response to a percent change in X.

Let "R" be the new parameter. We require that:

$$\frac{\Delta R}{R} = \frac{\Delta a}{a} \cdot S \qquad (3.92)$$

where S = scaling factor, so that a one-percent change in "a" will correspond to a S(>1) percent change in "R". Integrating both sides, we obtain:

$$\ln(R) = S.\ln(a) \tag{3.93}$$

$$R = a^S \tag{3.94}$$

Hence,

$$a = R^{1/S} \tag{3.95}$$

So, the percolation equation becomes:

$$\text{Perc} = \begin{cases} [A + (\frac{UM - A}{R^{X/S}}).(LZDR)^X].\frac{UC}{UM} & \text{for all } LZDR < a \\ UC & \text{for all } LZDR > a \end{cases} \tag{3.96}$$

Using the scaling factor S = 5, the R - X contour map of SLS criterion function values was obtained (Figure 3.19). [Note: nominal value $R = a^5 = 0.0028$]. The improvement in contour shapes is dramatic, and the use of a higher scaling factor is likely to result in even more circular contours.

Prior to implementation of a search routine, the value of the scaling factor "S" giving "optimal" contour shapes is, of course, unknown. Hence, a nominal value should first be used and, when the search has terminated, the relative sensitivities of "R" and "X" tested. If "R" is much more sensitive than "X", then the scaling factor should be increased and an attempt made to continue the optimization. Alternatively, the testing could be built into the search routine and carried out at prespecified occasions during the optimization, such as whenever a decision to reduce the step sizes of the parameters is made.

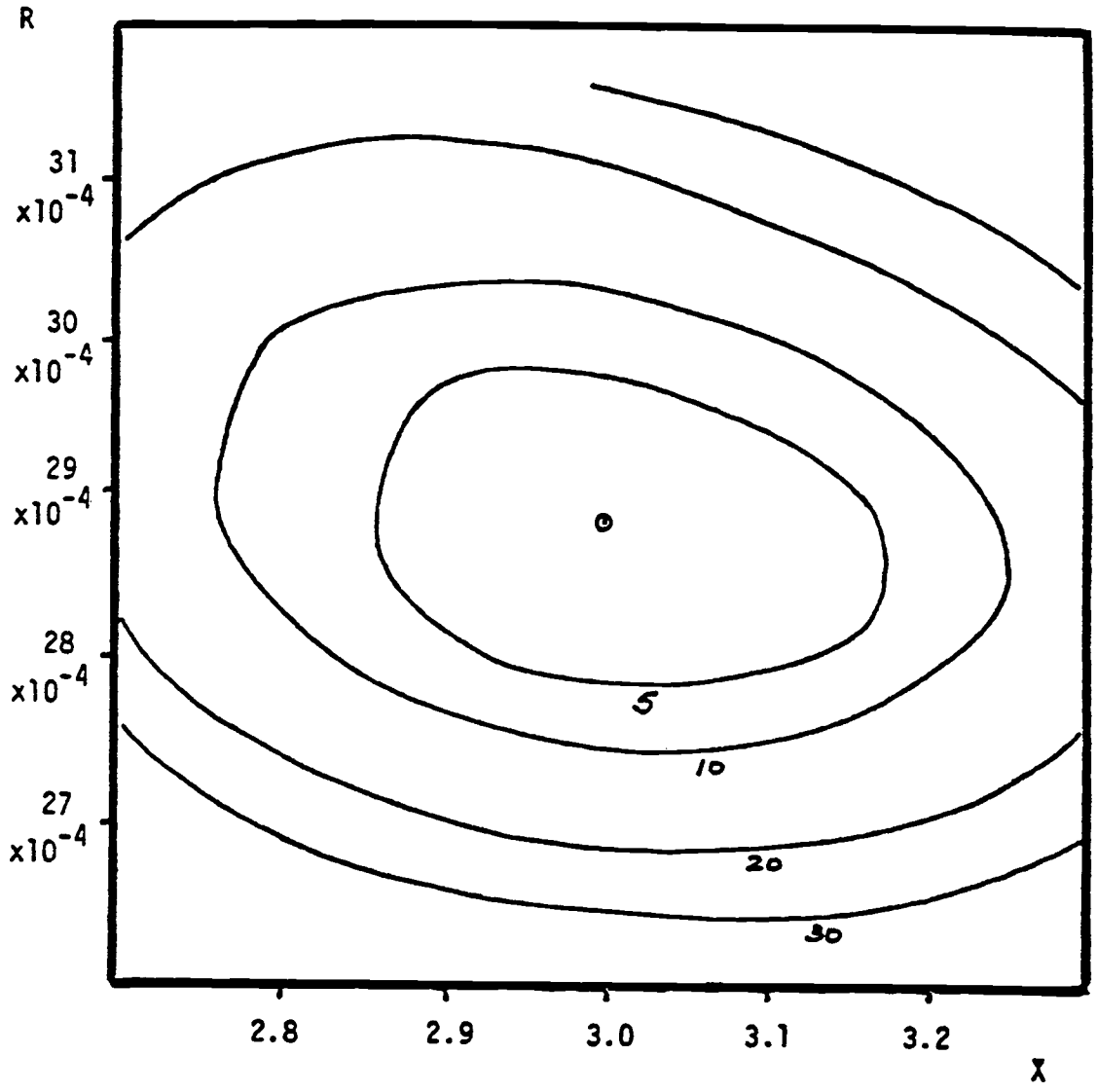


Figure 3.19. SLS contour map of R - versus - X for nominal $R = 29 \times 10^{-4}$ ($a = 0.31$ and $X = 3.0$)

Hence, it has been shown that appropriate reparameterization and scaling can be effectively used to obtain a better conditioned response surface which is more suitable for the application of direct search algorithms. It seems reasonable to expect that such an algorithm will have far less difficulty in searching for the optimum and is less likely to terminate prematurely. Also, the likelihood of obtaining unique parameter sets is enhanced, since at least one major cause of premature algorithm termination has been removed.

3.7.7 Experimental Testing of the Reparameterized Percolation Equation

3.7.7.1 Structural Identifiability Analysis

Using the data set RAIN-1 and the original parameterization (identified as P1), the structural identifiability of the model SIXPAR was computed. The condition number of the matrix and the index of multiparameter interaction were:

$$CN = 1.8 \times 10^{9} \quad (LN = 42{,}000)$$

$$MPI = 2.8 \times 10^{-4}$$

The two parameter concentricity indices and iteraction indices (Sorooshian and Arfi, 1982) are presented in Tables 3.4 and 3.5. Also, the sensitivity ratio for each of the parameters, along with the parameter confidence intervals and marginal confidence intervals for the hypothetical case of Gaussian output error (as described in Chapters 3 and 5), are presented in Table 3.6. Finally, the parameter correlation matrix for the case of Gaussian output error is presented in Table 3.7. The results clearly show that the percolation equation parameters Z and X are nearly non-identifiable. The condition

TABLE 3.4

Concentricity Indices in the 2-Parameter Subspaces for Parameterization P1 (0 = worst, 1 = best)

UM	•					
UK	0.04	•				
BM	0.44	0.04	•			
BK	0.01	0.41	10^{-3}	•		
Z	0.01	10^{-4}	0.02	10^{-4}		
X	0.73	0.04	0.79	0.02	10^{-3}	•
	UM	UK	BM	BK	Z	X

TABLE 3.5

Interaction Indices in the 2-Parameter Subspaces for Parameterization P1 (1 = worst, 0 = best)

UM	•					
UK	-0.01	•				
BM	-0.72	0.00	•			
BK	-0.02	-0.12	-0.02	•		
Z	0.00	0.00	0.00	0.00	•	
X	-0.42	0.00	0.86	0.00	0.02	•
	UM	UK	BM	BK	Z	X

TABLE 3.6

Indices of Marginal Confidence (MCI_i) and Confidence (CI_i) at the 95% Level and the Sensitivity (n_i) for Each Parameter of P1

Parameter	True Value	MCI_i (95%)	CI_i (95%)	n_i
UM	10.0	±0.819 σ	±3.991 σ	4.874
UK	0.5	±0.040 σ	±0.134 σ	3.393
BM	20.0	±1.103 σ	±5.692 σ	5.160
BK	0.2	±0.017 σ	±0.076 σ	4.500
Z	50.0	±63.154 σ	±1257.7 σ	19.914
X	3.0	±1.050 σ	±21.171 σ	20.162

σ = standard deviation of output error

TABLE 3.7

Parameter Correlation Matrix for Parameterization P1

UM	1.0					
UK	-0.83	1.0				
BM	-0.48	0.49	1.0			
BK	-0.17	-0.01	-0.71	1.0		
Z	-0.02	0.04	0.17	-0.24	1.0	
X	-0.09	0.11	0.25	-0.26	0.99	1.0
	UM	UK	BM	BK	Z	X

number CN is rather large, while multiparameter interaction as indicated by MPI is quite strong. From Table 3.6, we see that both Z and X have very poor precision. Further, while the precision ratios for parameters UM through BK are of the value of about 4, the precision ratios for Z and X are both around 20, indicating that these two parameters are the major contributors to parameter interaction. This is further supported by the correlation index between Z and X, which is 0.99, indicating near-deterministic interaction between the two. It is interesting to note that concentricity and interaction matrices are, in this case, not as useful as the other indices in detecting the poorly identifiable parameters.

These results seem to indicate that the problem of structural identifiability is essentially related to the percolation equation parameters and that the remaining parameters are properly identifiable. So, the next step in the analysis was to evaluate the proposed reparameterization of the percolation equation.

The new parameterization will be identified as P2. The structural identifiability matrix of the reparameterized model SIXPAR was computed. The condition number of the new matrix and the new index of multiparameter interaction were:

$$CN = 5.0 \times 10^{5} \quad (LN = 700)$$

$$MPI = 2.25 \times 10^{-2}$$

The value of the condition number has dramatically improved (four orders of magnitude), indicating a 60-fold reduction in the ratio of largest to smallest axis of the hyperellipse. Correspondingly, the multiparameter interaction has reduced (improved) by two orders of magnitude. The other identifiability indices are presented in Tables 3.8 through 3.11. The

TABLE 3.8

Concentricity Indices in 2-Parameter Subspaces for Parameterization P2 (0 = worst, 1 = best)

UM	•					
UK	0.04	•				
BM	0.41	0.04	•			
BK	0.01	0.41	10^{-3}	•		
Z	0.16	0.31	0.12	0.13	•	
X	0.08	10^{-3}	0.10	10^{-3}	0.01	•
	UM	UK	BM	BK	Z	X

TABLE 3.9

Interaction Indices in 2-Parameter Subspaces for Parameterization P2 (1 = worst, 0 = best)

UM	•					
UK	-0.04	•				
BM	-0.73	0.00	•			
BK	-0.02	0.12	0.02	•		
Z	-0.02	0.00	0.02	0.00	•	
X	0.00	0.00	0.00	0.00	0.00	•
	UM	UK	BM	BK	Z	X

TABLE 3.10

Indices of Marginal Confidence (MCI_j) and Confidence (CI_j) at the 95% Level and the Precision Ratio (n_j) for Each Parameter of P2

Parameter	True Value	MCI_j (95%)	CI_j (95%)	n_j
UM	10.0	±0.831 σ	±3.991 σ	4.803
UK	0.5	±0.040 σ	±0.134 σ	3.350
BM	20.0	±1.119 σ	±5.692 σ	5.087
BK	0.2	±0.017 σ	±0.076 σ	4.471
Z	0.31	±0.131 σ	±0.283 σ	2.160
X	3.0	±10.806 σ	±21.171 σ	1.959

σ = standard deviation of output error

TABLE 3.11

Parameter Correlation Matrix for Parameterization P2

UM						
UK	-0.83	1.0				
BM	-0.48	0.49	1.0			
BK	-0.16	-0.01	-0.71	1.0		
Z	-0.34	0.33	0.50	-0.29	1.0	
X	-0.09	0.11	0.25	0.26	-0.06	1.0
	UM	UK	BM	BK	Z	X

reparameterized model is clearly better structurally identifiable than the original model version. The confidence intervals indicate that parameter A can be much more precisely determined than parameter Z. The A-X parameter correlation is almost zero and there are no extremely high correlations. The precision ratios for parameters UM through BK have remained approximately the same from P1 to P2, but have improved considerably for the remaining two parameters. The interaction indices have, by and large, improved as well. However, some of the two-parameter subspace elongation indices have deteriorated a bit. One final observation is that the parameter X seems to be relatively much less sensitive than the other model parameters. It is possible that the sensitivity of parameter X might be improved if a data set with different characteristics is used.

3.7.7.2 Parameter Estimation

First, the original model parameterization (UM, UK, BM, BK, Z, X) identified as P1 was compared to the new parameterization (UM, UK, BM, BK, A, X) identified here as P2. The parameters UM through BK were started at their "true" values, while parameters Z and X (A and X) were started at 45.0 and 3.5 (0.3784 and 3.5), respectively. The results of these runs are reported in Table 3.12. As the structural identifiability analysis of Section 3.7.7.1 predicted, the algorithm was unable to recover the true parameters using the P1 parameterization, while it converged rather rapidly to the true values using the P2 parameterization. Clearly, parameterization P2 is superior to parameterization P1.

TABLE 3.12

Algorithm Convergence Results Comparing Parameterizations P1 and P2

Parameterization P1

Parameters	UM	UK	BM	BK	Z	X	Function	Intervals
Initial values	10.0000	0.50000	20.0000	0.2000	45.0000	3.5000	0.33	
Final values	9.6351	0.5089	19.7667	0.2080	45.0099	3.0021	0.01	18
Final % error	-3.65	+1.78	+1.17	+4.00	-9.98	+0.07		

Parameterization P2

Parameters	UM	UK	BM	BK	Z	X	Function	Intervals
Initial values	10.0000	0.5000	20.0000	0.2000	0.3784	3.5000	0.33	
Final values	10.0000	0.5000	20.0000	0.2000	0.3107	3.0000	10^{-7}	5
Final % error	0.0	0.0	0.0	0.0	0.0	0.0		

CHAPTER 4

THE STOCHASTIC PARAMETER ESTIMATION PROCEDURE

4.1 Introduction

In Section 1.2, we began by describing the identification problem as one of identifying a model M of a physical process $\mathscr{P}$ (e.g., the watershed rainfall-runoff process) for which certain characteristics such as inputs $\underline{u}$ (e.g., precipitation) and outputs $\underline{z}$ (e.g., streamflow, evapotranspiration) can be observed. We shall assume that a suitable model set $M(\underline{\theta})$ has been selected and that the chosen model set is identifiable (see Chapter 3 for a detailed discussion of this issue). Therefore, we are faced with the problem of finding those parameter values $\underline{\theta}^* \in \Theta$ for which the model $M(\underline{\theta}^*)$ is "closest" to the observed process $\mathscr{P}$. In order to do this, we must specify a quantitative measure of closeness which can be used as a basis for the comparison of alternative parameter values.

As indicated by the geometric interpretation of Figure 1.1, we can distinguish between two possible cases:

$$\text{Case 1:} \quad \mathscr{P} \in \{M(\underline{\theta}),\ \underline{\theta} \in \Theta\} \tag{4.1}$$

$$\text{Case 2:} \quad \mathscr{P} \notin \{M(\underline{\theta}),\ \underline{\theta} \in \Theta\} \tag{4.2}$$

Clearly, the watershed modeling problem and most other problems of practical interest are associated with Case 2. As pointed out by Baram and Sandell (1978), however, most model identification techniques rely on the assumption that the model set under examination contains the actual process $\mathscr{P}$. If this is indeed the case, then it is acceptable to consider as a measure of closeness of the model $M(\underline{\theta})$ from the process $\mathscr{P}$ ($\mathscr{P} = M(\underline{\theta}$ true)) the

distance between the parameter set $\underline{\theta}$ and the "true" parameter values $\underline{\theta}$ "true". Since, in practice, the model set $M(\underline{\theta})$ will usually not contain the process $\mathscr{P}$, there will not exist any such object as a "true" parameter value on which to base our comparison. Hence, we require some alternative method of comparing the models to the process. The most common and practical approach in this case is to compare the output behavior of the model to that of the process under similar input conditions. The "best" parameter set is then deemed to be that for which the model output is closest to the process output.

Figure 4.1 contains a simple visualization of this concept. Let u_t and z_t represent the "true" process inputs and outputs, respectively. It is assumed that the inputs and outputs are measured at discrete time intervals ($t = 1, \ldots n$). The model $M(\underline{\theta})$ operates on the input sequence to generate a set of model outputs $z_t(\underline{\theta})$ (the dependence on inputs $\underline{u}$ has been dropped for clarity of presentation) according to:

$$z_t(\underline{\theta}) = M(\underline{\theta}, u(t)) \quad \forall\, t = 1, \ldots n \tag{4.3}$$

where

$$u(t) = \{u_j,\ j = 1, \ldots t\} \tag{4.4}$$

Let us define the model output error ($m_t(\underline{\theta})$) as:

$$m_t(\underline{\theta}) = z_t(\underline{\theta}) - z_t,\ \forall\, t = 1, \ldots n \tag{4.5}$$

Clearly, we would like to ensure that the value of model output error is close to zero at all times. In other words, the model $M(\theta)$ is considered "closest" to the process $\mathscr{P}$ when the difference between its outputs $z_t(\theta)$

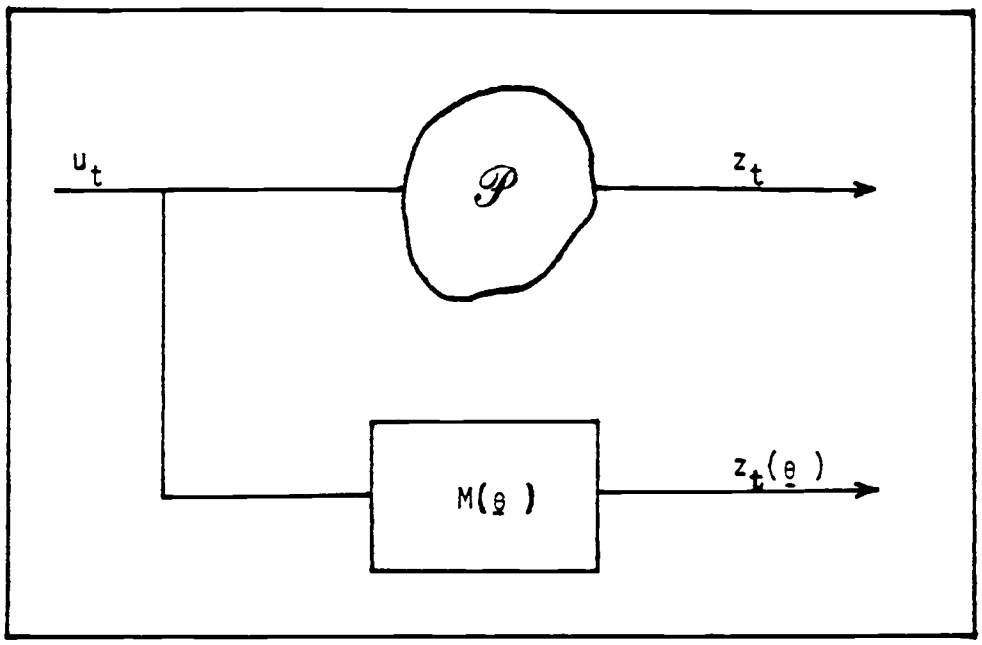


Figure 4.1. Representation of the process by a model

and the true outputs z_t is minimal. Hence, we are required to solve the problem:

$$\min_{\underline{\theta}\in\Theta} \quad \{|m_1(\underline{\theta})|, \ldots\ldots |m_n(\underline{\theta})|\} \tag{4.6}$$

Note that this is a vector optimization problem which is rather difficult to solve, especially if the number of parameters and the number of output values to be compared is large. In practice, therefore, we convert the problem into a simpler one by aggregating over all of the model errors to define a scalar measure of closeness F. The problem, therefore, becomes:

$$\min_{\underline{\theta}\in\Theta} \quad F = F(m_1(\underline{\theta}), \ldots. m_n(\underline{\theta})) = F(\underline{m}(\underline{\theta})) \tag{4.7}$$

where F is some (nonlinear) function of the model errors $m(\underline{\theta}) = \{m_1(\underline{\theta}), \ldots\ldots m_n(\underline{\theta})\}$.

Now, by specifying F as a measure of closeness of the model to the process, we will have converted the identification problem into an optimization (minimization) problem. However, we must ensure that the function F is carefully chosen so as to result in a well-defined optimization problem. Clearly, there are many possible functions that might be chosen for F. There are, however, no clear guidelines at this point on how F should be chosen.

It should be noted here that the model error $\underline{m}(\underline{\theta})$ arises due to two possible reasons:

(1) the model set $M(\underline{\theta})$ does not contain the process; i.e., the model structural equations are incorrect;

(2) the parameter set $\underline{\theta}$ is different from its optimal value.

In the ideal case, of course, there would be no model structural error to contend with, and the model errors could be driven to zero by an appropriate choice for $\underline{\theta}$. We would, therefore, like the chosen function F to have two important properties. First, it should take on its minimum possible value (F_{min}) if the model errors m_t, t = 1, ... n are all zero. Second, it should have a unique optimum (minimum) at the "best" parameter values $\underline{\theta}^*$. Clearly, if the model contains structural errors, then we will be unable to find a $\underline{\theta}$ such that the model output errors are all zero, and so the minimum value of F will be greater than F_{min}.

Conditions on F that will ensure that the function F has a unique minimum in the parameter space are rather difficult to obtain. A sufficient (though not necessary and, hence, restrictive) condition is that the function F is strictly convex in the parameter space. If the second derivative matrix $\nabla^2_\theta F$ exists, this condition is equivalent to requiring that the matrix $\nabla^2_\theta F$ is positive definite everywhere in the feasible parameter space. In general, for an arbitrary nonlinear (in the parameters) model, this is very difficult to ensure, since it depends on the particular structure chosen for the model set. Furthermore, we would like the condition for selection of F to be independent of the particular model structural equation. Using the chain rule for differentiation, we can show that if F and $\underline{m}(\underline{\theta})$ are twice differentiable,

$$\nabla^2_\theta F = \nabla_\theta \underline{m}^T(\underline{\theta}) \cdot \nabla^2_m F \cdot \nabla_\theta \underline{m}(\underline{\theta}) + G \tag{4.8}$$

where

$$\{G\}_{ij} = \left[\frac{\partial^2 m_1}{\partial\theta_i \partial\theta_j}, \ldots\ldots, \frac{\partial^2 m_n}{\partial\theta_i \partial\theta_j}\right] \cdot \nabla_m F \tag{4.9}$$

Now, in order for $\nabla_\theta^2 F$ to be a positive definite matrix, we require that:

(1) $\nabla_\theta \underline{m}(\underline{\theta})$ be of full rank;

(2) $\nabla_m^2 F$ be a positive definite matrix; and

(3) G be a positive definite matrix or be small enough that it is dominated by $\nabla \underline{m}_\theta^T(\underline{\theta}) \cdot \nabla_m^2 F \cdot \nabla_\theta \underline{m}(\underline{\theta})$.

The first and third conditions are related essentially to the structural identifiability of the model (discussed in greater detail in Chapter 5). Therefore, the second condition is the one of interest here. It states that we should choose the function F from the set of functions that is convex in $\underline{m}$. The conditions for a function F to be "acceptable" are therefore:

(1) $F(\underline{m})$ has a unique minimum F_{min} at $\underline{m} = 0$; and

(2) $F(\underline{m})$ is convex in $\underline{m}$, or $\nabla_m^2 F$ is positive definite for all $\underline{m}$.

There are many functions which can be found to satisfy these requirements. One of the simplest classes of functions, and the one to which many commonly-used measures of closeness belong, is of the form:

$$F = \sum_{t=1}^{n} f_t(m_t) \tag{4.10}$$

where each $f_t(m_t)$ is convex in m_t, has value zero at $m_t = 0$, and is positive everywhere else. The popular simple least squares function is a member of this class with $f_t(m_t) = (m_t)^2$. It is easy to verify that $\nabla_m^2 F = 2I$ and that $F(\underline{m})$ is positive everywhere except at $\underline{m} = 0$, where it assumes its minimum value of zero. It is important to note that if the model set $M(\underline{\theta})$ contains the process $\mathcal{P}$, then any "acceptable" function that might be used will

lead us to choose the same "best" parameter values, i.e., $\underline{\theta}^* = \underline{\theta}$ true. In practice, however, since the model set will not contain the true process, different functions will identify different "best" parameter values, depending on the relative importance given to minimizing different terms in the model error sequence. Hence, great care must be exercised in the selection of the function based on a clear understanding of the properties of the model errors that we might consider desirable.

4.2 The Stochastic Nature of the Parameter Identification Problem

As emphasized by Clarke (1973) and Sorooshian (1978), the input-output data available for selection of the parameter values contain significant amounts of measurement errors. The parameter identification problem may therefore be described in the following manner, with reference to Figure 4.2.

Let $\underline{u} = \{u_1, \ldots u_n\}$ be a sequence of n input values (e.g., precipitation) to the process $\mathscr{P}$ which give rise to a sequence $\underline{z} = \{z_1, \ldots z_n\}$ of process outputs (e.g., streamflow, evapotranspiration, etc.). Measurements are made of the inputs and outputs. The measurement process introduces random errors into the recorded measurements according to some underlying probability laws. Let $\mathscr{P}_u$ and $\mathscr{P}_z$ be the input and output measurement error probability laws, respectively. Denote the input measurement error at time t by η_t so that the recorded input measurement is:

$$\tilde{u}_t = u_t + \eta_t \tag{4.11}$$

Similarly, the output measurement error is denoted by e_t so that the recorded output measurement is:

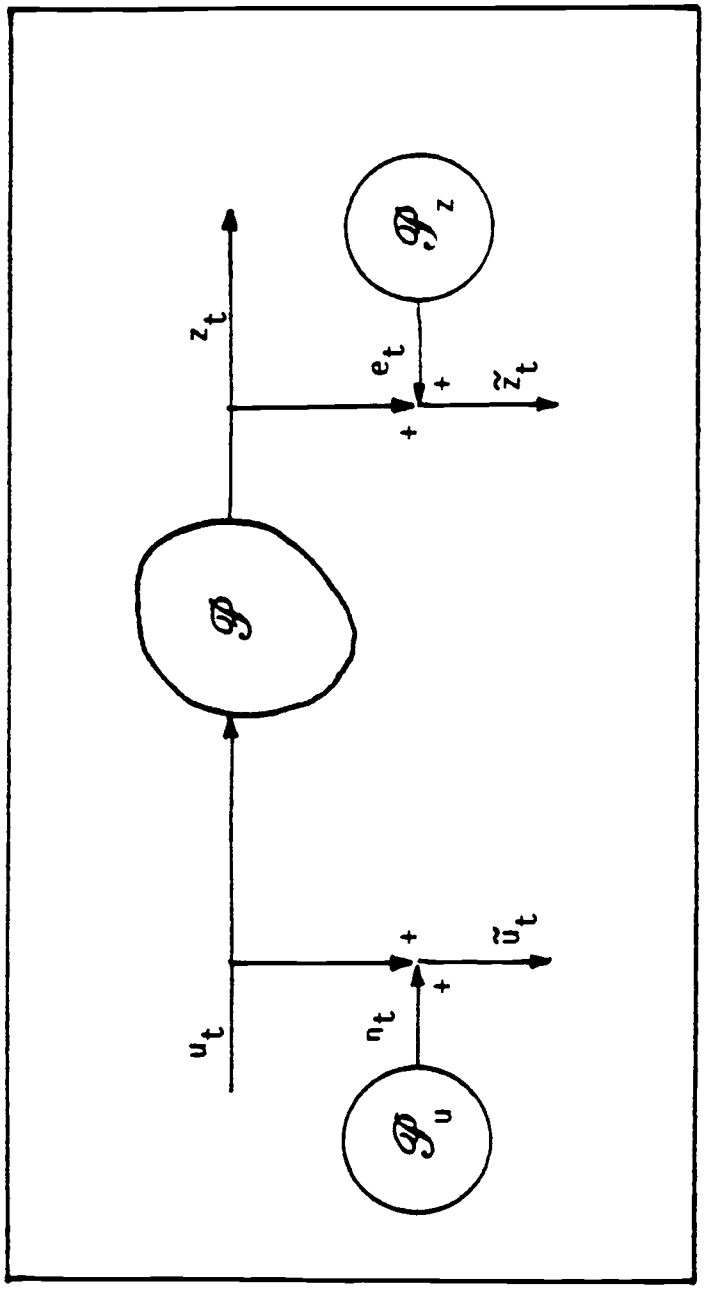


Figure 4.2. Errors introduced into data through the measurement process

$$\tilde{z}_t = z_t + e_t \tag{4.12}$$

Hence, the data available to us for the purposes of identification consist of:

(1) qualitative information about the watershed process;

(2) qualitative information about the measurement process; and

(3) quantitative error-corrupted input and output data.

Now since the output data are corrupted with errors, we no longer have available to us the "true" values of the outputs. Let us define the model residual to be the difference between the model output and the measured output:

$$r_t(\underline{\theta}) = z_t(\underline{\theta}) - \tilde{z}_t \qquad \forall\ t \tag{4.13}$$

Adding and subtracting z_t on the right-hand side and using Equations (3.5) and (3.12), we get:

$$r_t(\underline{\theta}) = z_t(\underline{\theta}) - z_t) - (\tilde{z}_t - z_t) \qquad \forall\ t \tag{4.14}$$

$$= m_t(\underline{\theta}) - e_t \qquad \forall\ t \tag{4.15}$$

Now, as discussed in Section 4.1, the objective of the identification problem is to find the $\underline{\theta}$ value for which the model errors $m_t(\underline{\theta})$ are "close" to zero. However, due to the presence of measurement error, the model error is <u>not</u> directly observable. We must, therefore, find some method which allows us to manipulate the residuals (which are observable) with the effect of driving the model errors close to zero. We note once again that since in

practice the model set does not contain the process, we shall not be able to drive the model errors completely to zero. It is important to bear both of the above factors in mind when selecting a measure of closeness.

4.3 Parameter Identification Methodologies Available

As emphasized in the previous section, the chosen measure of closeness should, if possible, reflect the a priori knowledge (or lack of it) that we have about the system under study. There are two basic elements which we have to consider in the parameter identification process. They are:

(1) Nature of the model structure--does the model relate inputs to outputs in a deterministic or a probabilistic manner, and does the model set contain the process; and

(2) Are the input-output data accurate enough to be considered deterministic, or do they contain significant measurement errors.

The major methodologies developed by the scientific community to handle various cases are indicated in Table 4.1 (adapted from Eykhoff, 1982). The level of computational difficulty involved in the implementation of these methodologies increases down the list. The use of the SLS and other subjectively-chosen measures of closeness for the identification of conceptual rainfall runoff models is indicated by Case 1. The approach taken by Sorooshian (1978) and Sorooshian and Dracup (1980) fits into Case 2. The work of Goldstein and Larimore (1980) and Posada and Bras (1982) fits into Case 3. To the knowledge of this author, no application of Case 4 has been reported.

TABLE 4.1

Measures of Closeness Associated with Different Kinds of Models and Data

Case	I/O Data	Model Structure	Model Parameters	Appropriate Measure of Closeness
I	Deterministic	Deterministic	Deterministic	Any "acceptable" function
II	Stochastic	Deterministic	Deterministic	Likelihood function
III	Stochastic	Stochastic	Deterministic	Likelihood function
IV	Stochastic	Stochastic	Stochastic	Bayes function

Table 4.1 indicates that for models wherein the parameters are deterministic unknown constants, the method of maximum likelihood is a very powerful approach. Furthermore, as Edwards (1972) and Baram and Sandell (1978) point out, the maximum likelihood procedure is the only methodology so far available in systems theory which does not require the assumption that the process be a member of the model set. It does, however, require us to make some assumptions about the nature of the underlying measurement error probability laws.

It is important to point out here that in the application of the Case 3 approach wherein the model structure is considered to be stochastic, the exact maximum likelihood approach is, to date, only practically feasible for linear models with Gaussian structural error. In the work of Goldstein and Larimore (1980) and Posada and Bras (1982), an approximate maximum likelihood procedure was applied based on the linearization approach of the extended Kalman filter (Kitanidis and Bras, 1978). Although the incorporation of the assumption that the data are stochastic has been shown to result in significant improvements over the usual SLS approach by Sorooshian and co-workers, the marginal benefit of adopting the more computationally demanding Case 3 approach has not, in the opinion of this author, been successfully demonstrated. In this work, therefore, the emphasis will be laid on improving on the Case 2 procedures currently available for conceptual rainfall-runoff model parameter identification.

4.4 The Method of Maximum Likelihood

Consider a process that is completely described by a probability law $\mathscr{P}_X$, which generates objects X that are random variables. Our object is to identify a model $P(X|\theta)$ parameterized by $\underline{\theta}$ of the process $\mathscr{P}_X$. The

information which is available consists of a data set $D = \{x_1, \ldots\ldots x_n\}$ of n independent observations of the objects X. This set D is a particular realization of the possible data sets that might have been realized due to the fact that X is a random variable. The practical identification problem is to determine that model $P(X|\theta)$ which most likely gave rise to the observed data set, i.e., we would like the model to be capable of generating objects, say "Y", that are closest to being "statistically identical" to the observed objects "X" (two objects are defined to be statistically identical if they obey the same probability law). The method of maximum likelihood pursues this objective by maximizing over the space of all models (for instance, over a feasible model set) the "likelihood" that the model-generated objects "Y" are statistically identical to the observed objects "X". In other words (Baram and Sandell, 1978), given a model set $P(X|\theta)$, we choose the parameters $\underline{\theta}^*$ such that the information in the observations for distinguishing between the model $P(X|\theta^*)$ and the process $\mathscr{P}_x$ is minimal.

Therefore, we would like to choose the measure of closeness in such a way that it makes the most use of the information contained in the available data. The function which does this is described in the statistical literature as the "likelihood function", according to the following proposition (Edwards, 1972):

> Proposition (the likelihood principle): Within the framework of a statistical model, all of the information which the data provide regarding the relative merits of differential hypotheses (e.g., different parameter sets) is contained in the likelihood function of those hypotheses given the data.

The form of the likelihood function is described by the following definition:

> Definition: The likelihood function $\mathcal{L}(\theta|x)$ of a hypothesis given an observation x and a specified model $P(X|\theta)$ of the probability law $\mathcal{P}_x$ is proportional to $P(X|\theta)$, the constant of proportionality, "K" being arbitrary.

Hence, the likelihood function is given by:

$$\mathcal{L}(\underline{\theta}|x) = K \cdot P(X|\underline{\theta}) \Big|_{X = x} \tag{4.16}$$

Note that whereas $P(X|\theta)$ is a function of the variable X, $\mathcal{L}(\underline{\theta}|x)$ is a function of the variable $\underline{\theta}$ to be determined. The likelihood function is therefore a measure of the information contained in the observation x about the value of the parameter $\underline{\theta}$.

Now, the data set $D = \{x_1 \ldots x_n\}$ consists of n observations of the random variable X, their joint distribution being given by $P(D|\theta)$. If the observations are independently distributed, then:

$$P(D|\underline{\theta}) = \prod_{i=1}^{n} P_i(x_i|\underline{\theta}) \tag{4.17}$$

and hence the likelihood function is:

$$\mathcal{L}(\underline{\theta}|D) = \prod_{i=1}^{n} \mathcal{L}_i(\underline{\theta}|x_i) \tag{4.18}$$

In order that the information about the parameters $\underline{\theta}$ be additive over the observations, it is common to work instead with the natural logarithm of the likelihood function, i.e.,

$$L(\underline{\theta}|D) = \ln \mathcal{L}(\underline{\theta}|D) \tag{4.19}$$

$$= \sum_{i=1}^{n} \ln \mathcal{L}(\underline{\theta}|x_i) \tag{4.20}$$

$$= \sum_{i=1}^{n} \ln P_i(x_i|\underline{\theta}) + \text{constant} \tag{4.21}$$

Here, $L(\underline{\theta}|D)$ is the information available about the parameters $\underline{\theta}$ contained in the finite number of observations contained in the observation sequence D. In order to condition our choice of $\underline{\theta}$ on the entire sample space of X and not just on a particular observation sequence, we can define the expected supported for the hypothesis $\underline{\theta}$ by:

$$S(\mathcal{P}_X; L(\underline{\theta}|x)) = E_X\{L(\underline{\theta}|x)\} \tag{4.22}$$

$$\int_X L(\ |x) \cdot \mathcal{P}_X(x) \cdot (dx) \tag{4.23}$$

where X is the feasible space for x, and E_X is the expectation operator over X. Substituting for $L(\underline{\theta}|x)$ from Equations (4.16) and (4.19), we get:

$$S(\mathcal{P}_X; L(\underline{\theta}|\cdot)) = \int_X \ln P(x|\underline{\theta})\ \mathcal{P}_X(x)\ dx + \text{constant} \tag{4.24}$$

as the measure of support for the parameter set $\underline{\theta}$.

Using Equation (4.24), the Kullbach-Leibler measure of mean information (Akaike, 1974) is defined by:

$$\mathcal{I}[\mathcal{P}_X; P(\cdot|\theta)] = S[\mathcal{P}_X; \mathcal{P}_X] - S[\mathcal{P}_X; P(\cdot|\underline{\theta})] \tag{4.25}$$

$$= \int_X [\ln \mathcal{P}_X(x) - \ln P(x|\theta)]\ \mathcal{P}_X(x)\ dx \tag{4.26}$$

Note that $\mathcal{I}[\cdot;\cdot]$ is always greater than zero unless the model is identical to $\mathcal{P}_X$ over the entire space X, in which case it is equal to zero. Hence, $P(x|\underline{\theta})$ will be closest to $\mathcal{P}_X$ (in the expected information sense) if $\mathcal{I}(\cdot;\cdot)$ is minimum, or hence $S[\mathcal{P}_X; P(\cdot|\theta)]$ is a maximum. An arbitrarily close approximation can be computed for the expected support function without knowing the actual process $\mathcal{P}_X$, as follows (Akaike, 1974):

$$\bar{S} = \frac{1}{N} \sum_{L=1}^{N} \ell n\, P_i(x_i|\underline{\theta}) + \text{constant} = \frac{1}{N} L(\underline{\theta}|D) + \text{constant} \tag{4.27}$$

where

$$\lim_{N \to \infty} \bar{S} = S(\mathcal{P}_X; P(\cdot|\underline{\theta})) \tag{4.28}$$

Note that maximization of $\bar{S}$ is equivalent to maximization of the likelihood function $L(\underline{\theta}|D)$. Hence, solution of the problem

$$\max_{\theta \in \Theta} L(\underline{\theta}|D) = \sum_{L=1}^{N} \ell n\, P_i(x_i|\underline{\theta}) + \text{constant} \tag{4.29}$$

will give us the closest model $P(x|\underline{\theta})$ to the process $\mathcal{P}_X$ conditional on the chosen model set and the available set D of random variable observations.

4.5 Likelihood and the Identification of Dynamic Models

4.5.1 Introduction

The objective of the likelihood approach is to infer which of the available models "most likely" gave rise to the observed random variables,

i.e., it chooses the model (parameters) that is (are) best supported by the available information. In Section 4.3, we indicated that the random variables in the watershed modeling problem consisted of the input and output errors as described by Equations (4.11) and (4.12), respectively. Now, the measurement consists of two parts, the first being a deterministic time varying quantity representing the true input or output, which we shall assume is associated with a deterministic physical process (the watershed), and the second being a random quantity associated with a probabilistic process. Hence, the object we must model consists of two parts: (1) the process (watershed) model, and (2) the probability law model of the measurements, since both of these parts taken together are required to describe the probability distribution of the measurements. From Equations (4.11) and (4.12), we obtain:

$$\mathscr{P}_{\eta}(\eta) = \mathscr{P}_{\eta}(\tilde{\underline{u}} - \underline{u}) \tag{4.30}$$

and

$$\mathscr{P}_{e}(\underline{e}) = \mathscr{P}_{e}(\tilde{\underline{z}} - \underline{z}) \tag{4.31}$$

In practice, we will not know the true probability laws and will model them by some suitable parametric probability model as:

$$P_{\eta}(\underline{\eta}|\psi_u) = P_{\underline{\eta}}([\tilde{\underline{u}} - \underline{u}]|\underline{\psi}_u) \tag{4.32}$$

and

$$P_{e}(e|\psi_z) = P_{\underline{\eta}}([\tilde{\underline{z}} - \underline{z}]|\underline{\psi}_z) \tag{4.33}$$

where $\underline{\psi}_u$ and $\underline{\psi}_z$ are parameters of the probability models. Also, the quantities $\underline{u}$ and $\underline{z}$ are assumed to be generated by some deterministic underlying processes parameterized by $\underline{\theta}_u$ and $\underline{\theta}$, respectively. Hence, the complete models of the distributions of the measurements are given by:

$$P_{\underline{\eta}}(\underline{\eta}|\underline{\psi}_u,\underline{\theta}_u) = P_{\underline{\eta}}(\tilde{\underline{u}} - \underline{u}(\underline{\theta}_u)|\underline{\psi}_u,\underline{\theta}_u) \tag{4.34}$$

and

$$P_{\underline{e}}(\underline{e}|\underline{\psi}_z,\underline{\theta}_u\theta) = P_{\underline{e}}(\tilde{\underline{z}} - \underline{z}(\underline{\theta},\underline{\theta}_u)|\underline{\psi}_z,\underline{\theta}_u) \tag{4.35}$$

Note that the model of the output error depends on the parameters of deterministic input process through the input-output process model relationship (Equation 4.3).

To derive the likelihood function, we require the joint distribution of the input and output errors. In the case of the watershed model, it is reasonable to assume that the input and output measurement errors are independent. In this case, the likelihood function to be maximized is given by (dropping the arbitrary constant):

$$\begin{aligned} L(\underline{\psi}_u,\underline{\psi}_z,\underline{\theta}_u,\underline{\theta}|D) = {} & \ell n\ P_{\underline{\eta}}([\tilde{\underline{u}} - \underline{u}(\underline{\theta}_u)]|\underline{\psi}_u,\underline{\theta}_u \\ & + \ell n\ P_{\underline{e}}([\tilde{\underline{z}} - \underline{z}(\underline{\theta},\underline{\theta}_u)]|\underline{\psi}_z,\underline{\theta},\underline{\theta}_u \end{aligned} \tag{4.36}$$

4.5.2 Input Measured Without Error

In the case that the input error $\underline{\eta}$ is zero or insignificant compared to the model and output errors, the likelihood equation to be maximized

simplifies to:

$$L(\psi_z, \underline{\theta}|D) = \ell n\ P_{\underline{e}}([\underline{\tilde{z}} - \underline{z}(\underline{\theta})]|\psi_z, \underline{\theta}) \tag{4.37}$$

Maximization of Equation (4.37) will result in the jointly identified watershed process and output error model that is best supported by the available data set D. It is easy to show that this also results in the watershed model for which the model error is closest to zero. Let the structure of the output error be given by:

$$e_t = \hat{e}(\underline{\psi}_z) + a_t \tag{4.38}$$

where $\hat{e}(\underline{\psi}_z)$ is the "most likely" value of the measurement error, and $\underline{a} = \{a_1, \ldots a_n\}$ is a sequence of independent identically distributed random variables. From Equation (4.38), the distribution of $\underline{a}$ is:

$$P_{\underline{a}}(\underline{a}|\psi_z, \underline{\theta}) = P_{\underline{a}}([\underline{\tilde{z}} - \underline{z}(\underline{\theta}) - \underline{\hat{e}}(\psi_z)]|\psi_z, \underline{\theta}) \tag{4.39}$$

Therefore, maximization of Equation (4.37) is identical to maximization of Equation (4.39) and, hence we may rewrite the likelihood function as:

$$L(\underline{\psi}_z, \underline{\theta}) = \ell n\ P_{\underline{a}}(\underline{\tilde{z}} - \underline{z}(\underline{\theta}) - \underline{\hat{e}}(\underline{\psi}_z)|\underline{\psi}_z, \underline{\theta}) \tag{4.40}$$

Now, from Equations (4.5), (4.12), and (4.38), we have:

$$\underline{\tilde{z}} - \underline{z}(\underline{\theta}) - \underline{\hat{e}}(\underline{\psi}_z) = -\underline{m}(\underline{\theta}) + \underline{a} \tag{4.41}$$

If the term $(\underline{\tilde{z}} - \underline{z}(\underline{\theta}) - \underline{\hat{e}}(\underline{\psi}_z))$ is made equal to $\underline{a}$, then from Equation (4.41),

we see that this implies that the model error $\underline{m}(\underline{\theta})$ is made zero. Hence, maximization of Equations (4.37) or (4.39) results in maximization of the likelihood that the model error is zero.

In order to completely specify the identification problem, the underlying structures for $\underline{z}(\underline{\theta})$ and $\hat{\underline{e}}(\underline{\psi}_z)$ need to be specified. The structure of $\underline{z}(\underline{\theta})$ is already given in Equation (4.3). In this report, it will be assumed that the error sequence e_t can be modeled as being generated by the perturbation of a nonlinear (or linear) model (often called a filter) by a white-noise sequence w_t, with most likely value zero, i.e.,

$$e_t = N(\underline{\psi}_z, e_{t-1}, e_{t-2}, \ldots, w_t) \tag{4.42}$$

Hence, the most likely value of e_t is given by:

$$\hat{e}_t(\psi) = N(\underline{\psi}_z, \hat{e}_{t-1}, \hat{e}_{t-2}, \ldots \hat{w}_t) \tag{4.43}$$

and since $\hat{w}_t$= zero:

$$e_t(\psi) = N(\psi_z, \hat{e}_{t-1}, \hat{e}_{t-2}, \ldots) \tag{4.44}$$

where

$$\hat{e}_{t-j} = \tilde{z}_{t-j} - z_{t-j}(\underline{\theta}) \tag{4.45}$$

The error structures used by Sorooshian (1978) and Fulton (1982) were members of this class, where the function $N(\cdot)$ was a linear equation.

The complete statement of the stochastic parameter identification problem is therefore:

$$\max_{\substack{\theta \in \Theta \\ \underline{\psi}_z \in \Psi_z}} \quad L(\underline{\theta}, \underline{\psi}_z | D) = \ell n \; P_a(\underline{a} | \underline{\psi}_z, \underline{\theta})$$

where $\hat{a}_t = \tilde{z}_t - z_t(\underline{\theta}) - \hat{e}_t(\psi_z)$ (4.46)

$$z_t(\theta) = M(\underline{\theta}, \underline{u}(t))$$

$$\hat{e}_t(\psi_z) = N(\underline{\psi}_z, \hat{e}_{t-1}, \ldots)$$

$$\hat{e}_{t-j} = \tilde{z}_{t-j} - z_{t-j}(\underline{\theta})$$

4.5.3 Inputs and Outputs Contain Error

The problem of identification when the inputs and outputs contain errors is significantly more difficult than the case when inputs are error free. Approaches to dealing with this case have been proposed by Gertler (1979), Otter (1981), and Soderstrom (1981), among others. The work of Gertler (1979) relates to closed loop identification and is applicable to control problems. The approach suggested by the latter two authors is to model the input sequence as the output of a linear or nonlinear model perturbed by white noise and to use a Kalman filter-type approach to identify the distribution of the input errors. This approach, however, requires that the process model be also treated as a stochastic model, and hence a joint state-parameter identification procedure is necessary. Goldstein and Larimore (1980) applied this technique to a stochastic version of the SMA-NWSRFS model. However, it was found (Larimore, 1981) that the model performance was better when the input was assumed to be deterministic. It is the opinion of this author that the problem of modeling the input is probably more difficult than the problem of modeling the watershed process and that

the possible gains in forecasting ability are significantly offset by the difficulty of solving the identification problems. Hence, it will be assumed throughout the rest of this work that the input error is insignificant in comparison to the output and model errors.

4.6 Discussion of the Likelihood Approaches Suggested by Sorooshian (1978) and Fulton (1982)

4.6.1 Introduction

Sorooshian (1978) and Fulton (1982) both assumed that the streamflow errors belonged to a Gaussian distribution with zero mean and unknown covariance. Sorooshian (1978) proposed that the errors be modeled by a linear autoregressive model of the form:

$$e_t = \rho\, e_{t-1} + a_t \tag{4.47}$$

where ρ is the first-lag autocorrelation parameter that indicates the degree of systematic error in the measurements, and a_t is a Gaussian, zero mean, uncorrelated sequence with Var $[a_t] = \sigma^2_{a_t}$. Using this basic structure, Sorooshian developed two different likelihood functions, the first called the AMLE (or MLE 1) for the case of correlated errors with homogeneous variance $[\sigma^2_t = \sigma^2 \forall t]$, and the second called the HMLE (or MLE 2) for the case of uncorrelated $[\rho = 0]$ inhomogeneous variance errors. These functions are contained in Appendix 6. Sorooshian and Dracup (1980) reported good performance of these procedures in simulation studies. An application of the procedures to the identification of the SMA-NWSRFS model indicated that the HMLE procedure was superior to the AMLE or to the commonly-used SLS approach (Sorooshian, Gupta, and Fulton, 1983). This indicated that heteroscedasticity (inhomogeneous variance of the errors) was a far more serious problem

than was error correlation. However, significant correlation was also present in the residual time series, indicating that both issues should be simultaneously addressed. Fulton (1982) proposed a procedure to handle both correlation and heteroscedasticity and also extended it to handle multiple-lag correlation. This analysis revealed some instabilities in the original HMLE procedure reported by Sorooshian (1978). The HMLE function (see Appendix 6) requires the estimation of a weight w_t to be given to each squared residual based on its expected variance where the weight takes the form:

$$w_t = \frac{\sigma^2}{\sigma_t^2} \cong [E\{\tilde{z}_t\}]^{2(\lambda-1)} \quad \forall \ t \tag{4.48}$$

where λ and σ^2 are parameters of the output error probability model. Under the Gaussian assumption, the best estimate of $E\{\tilde{z}_t\}$ is given by $z_t(\hat{\theta})$ where $\hat{\theta}$ are the maximum likelihood values of the parameters of the model. Sorooshian (1978) suggested the use of $z_t(\underline{\theta})$ in place of $E\{\tilde{z}_t\}$ at each iteration of the solution procedure. Fulton (1982) proposed divergence of this procedure when tested on the SMA-NWSRFS model and proposed instead that the measured output values $\tilde{z}_t$ be substituted for $E\{z_t\}$. The procedure was shown to be stable under these conditions (Sorooshian, Gupta, and Fulton, 1983; Ibbitt and Hutchinson, 1983). However, no analysis of the bias introduced by this assumption was presented.

4.6.2 Bias Caused by Fulton's (1982) Proposed Modification

The weights to be employed in the likelihood functions of Sorooshian (1978) and Fulton (1982) are given by:

$$W_t = (\bar{z}_t)^{2(\lambda-1)} \tag{4.49}$$

where $\bar{z}_t = E\{z_t\}$. Fulton proposed the use of the following weight estimate:

$$\tilde{W}_t = (\tilde{z}_t)^{2(\lambda-1)} \tag{4.50}$$

Now, $\tilde{z}_t$ is a Gaussian random variable with mean equal to $\bar{z}_t$ and variance equal to σ_t^2. Expanding Equation (4.50) about the point $(\bar{z}_t, w_t)$ in a Taylor series to second order, we obtain:

$$\tilde{W}_t - W_t \cong 2(\lambda-1)(\bar{z}_t)^{2\lambda-3}[\tilde{z}_t - \bar{z}_t]$$

$$+ (\lambda-1)(2\lambda-3)(\bar{z}_t)^{2\lambda-4}[\tilde{z}_t - \bar{z}_t]^2 + \ldots \tag{4.51}$$

Taking the expectation of both sides,

$$E\{\tilde{W}_t\} - W_t = (\lambda-1)(2\lambda-3)(\bar{z}_t)^{2\lambda-4}\sigma_t^2 \tag{4.52}$$

Now, substituting for σ_t^2 from Equation (4.48), we get:

$$E\{\tilde{W}_t\} = W_t + \frac{(\lambda-1)(2\lambda-3)\sigma^2}{(\bar{z}_t^2)} \tag{4.53}$$

The above expression indicates that the second order bias in the estimate of the weights obtained using the approach suggested by Fulton (1982) is positive and can be quite large for small flows, non-negligible variances, and for $\lambda \ll 1.0$. For example, when $\lambda = 0$, we obtain:

$$W_t = 1/(\bar{z}_t)^2 \tag{4.54}$$

and

$$E\{\tilde{W}_t\} = W_t + 3\sigma^2/(\bar{z}_t)^2 \tag{4.55}$$

Note that for $\sigma^2 = 0.58$, the bias is of the same order as the true value of the weight for all values of output level $\bar{z}_t$.

The above discussion is based on a simple second order analysis. A simulation experiment to verify this analysis and to establish its potential impact on the identification procedure was conducted as follows. A thousand values for z_t were generated uniformly on the range (30,60). Next, a sequence of Gaussian pseudo-random variables a_t was generated with zero mean and unit variance. Now, from Equation (4.48), we have:

$$\sigma_t^2 = (z_t)^{2(1-\lambda)}\sigma^2 \qquad \forall \quad t \tag{4.56}$$

and hence the following equation was employed to create the output errors with the correct variance property:

$$e_t = (z_t)^{(1-\lambda)} \cdot \sigma \cdot a_t \tag{4.57}$$

These were added to the true outputs to obtain:

$$\tilde{z}_t = z_t + (z_t)^{(1-\lambda)} \cdot \sigma \cdot a_t \tag{4.58}$$

The $\tilde{z}_t$ values obtained using Equation (4.58) were used to compute the weights

of the HMLE function, and the parameter λ of the probability model was estimated by maximization of the likelihood.

Two sets of experiments were conducted. In the first, the true value of λ was chosen to be 0.5 and in the second $\lambda = 0.1$. For each case, σ^2 was varied with values (0.005, 0.01, 0.05, 0.1, 0.2). Note that for $\lambda = 0.5$, this corresponds to σ_t^2 = 0.5%, 1,0%, 5.0%, 10%, and 20% of the true value. The results are displayed in Table 4.2. The bias in the estimate λ is clearly substantial. This result poses a dilemma for the person wishing to employ the procedures of Sorooshian (1978) or Fulton (1982). If $z_t(\theta)$ is used in computation of the weights, instabilities may occur. If $\hat{z}_t$ is employed instead, the parameter estimates are likely to be biased. In the next section, a new look is taken at the problem of representing the output error probability law for streamflow measurements.

4.7 An Alternative Likelihood Procedure for Watershed Models

4.7.1 Introduction

As discussed in Section 4.5, an important part of the identification procedure lies in the selection of a realistic model of the measurement error probability law. In the procedures proposed by Sorooshian (1978), it was assumed that the streamflow measurement error was Gaussian. However, a closer examination of the streamflow measurement process based on the rating curve reveals that this assumption is likely to be violated in practice.

The most common method for measuring streamflow in rivers or large streams is as follows (Linsley et al., 1975). The actual physical measurement made at the river cross section of interest is of the "river stage", i.e., the level of the river elevation above some fixed data. These

TABLE 4.2

λ Estimate for Different λ_{true} and σ^2 Values

σ^2	$\lambda_{true} = 0.5$	$\lambda_{true} = 0.1$
0.005	0.570	0.375
0.010	0.590	0.520
0.050	0.650	0.880
0.100	0.710	1.010
0.200	0.790	-

data are stored in some numerical, digital, or analog form by a recording device, and are converted into streamflow measurements by means of a calibrated function called a "rating curve" or a "stage-discharge" relation. For most measurement stations, a simple plot of stage versus discharge is satisfactory (see Figure 4.3). Such a curve is approximately parabolic but may show some irregularities if the flow regime changes between high and low flows, or if the cross section is irregular. The "rating curve" for a particular station must be carefully prepared in advance. It is constructed by observing stage levels and simultaneously recording the discharge at the station using current velocity meters and the area of the river cross section. These points are plotted on a graph and a simple curve fitted through them. Clearly, the fitted curve is only an approximation to the complex relationship between stage and discharge, and some error in fitting the relationship will exist.

This discussion indicates that there are at least two sources which introduce error into the streamflow data. The first source is the stage recording device that may inaccurately measure the river stage. The second source is the inaccuracy in the stage-discharge relationship. We must use this information to deduce the nature of the probability law of the streamflow measurements. Let us consider the recorded stage which is the physical quantity measured. For a reasonably calibrated measurement device, the error in each stage measurement is as likely to be positive as it is negative, i.e., the distribution of the stage measurement error is likely to be approximately symmetric. Further, it is reasonable to expect the most likely value of the error to be zero, unless some systematic error is introduced into the measurements due to poor instrument calibration. In any case, the distribution of the stage measurements is likely unimodal and

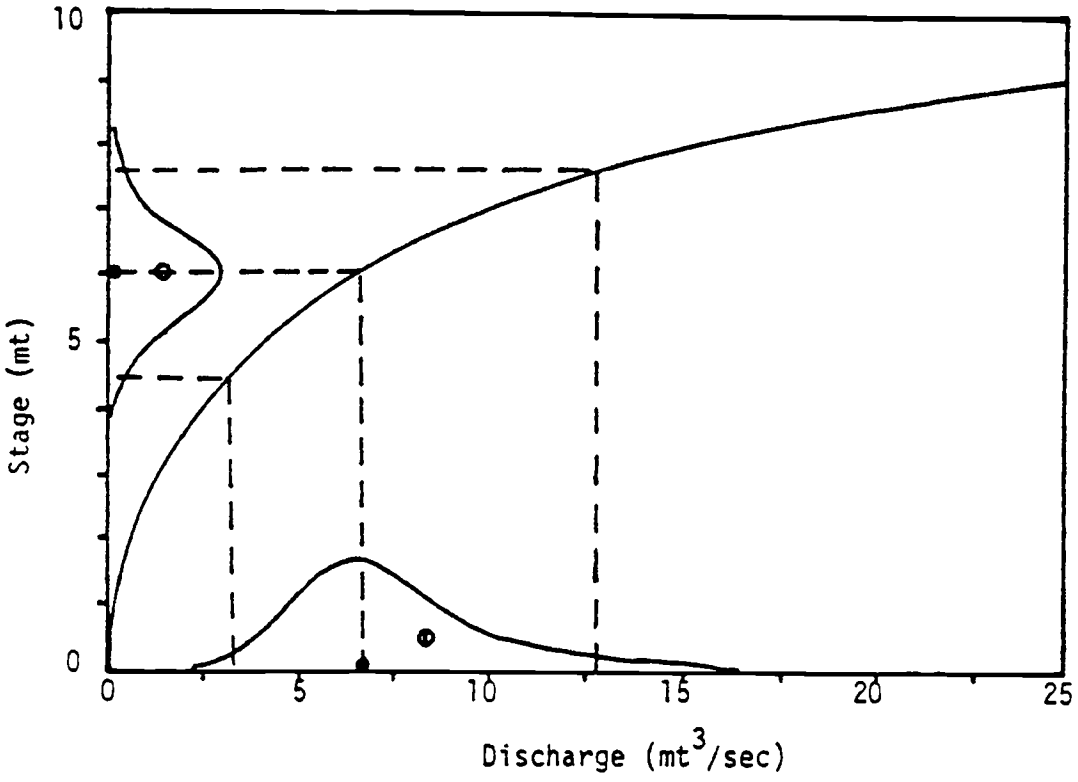


Figure 4.3. Typical stage-discharge relationship

⊙ - Expected value

● - True value (assuming no errors in curve calibration)

symmetric with mode close to zero. This stage measurement is converted into a streamflow measurement by means of the nonlinear rating curve (Figure 4.3). Clearly, due to the nonlinearity of the relationship, the probability distribution of the streamflow measurement errors will be positively skewed, and the expected streamflow measurement will be positively biased. This skew, or bias, will become more pronounced as the variance of the stage measurement error increases, and/or the nonlinearity of the rating curve becomes more pronounced. Thus, the assumption of Gaussian error in the streamflow measurement is probably unsatisfactory. It is interesting to note (Sorooshian, 1978) that as the stage increases, even if the variance of the stage error is a constant, the variance of the streamflow error increases, thus making high-stage streamflow data less reliable than low-stage streamflow data in general.

4.7.2 A Probability Law Model for Streamflow Measurements

Let the probability distribution of the stage measurements be given by $P_h(\underline{\tilde{h}})$ and the relationship between the streamflow measurement $(\tilde{z})_t$ and the stage measurement $(\tilde{h})_t$ be given by the invertible function:

$$\tilde{h}_t = R(\underline{\psi}_r, \tilde{z}_t) \tag{4.59}$$

The probability distribution $P_z(z)$ of the streamflow measurement can be derived using the theorem on transformation of random variables (see, e.g., Guttman et al., 1971, pp. 122-128; C.T. Hann, 1977, pp. 36-38):

$$P_z(\underline{\tilde{z}}) = P_h(\underline{\tilde{h}} = R(\psi_r; \underline{\tilde{z}})) \cdot |\, J(\underline{\tilde{h}}:\underline{\tilde{z}})\,| \tag{4.60}$$

where $J(\tilde{h}:\tilde{z})$ is the Jacobian of the transformation from $\underline{\tilde{h}}$ to $\underline{\tilde{z}}$ and is given by:

$$J(\tilde{h}:\tilde{z}) = \frac{\partial \underline{h}}{\partial \underline{z}} \tag{4.61}$$

Hence, we get:

$$P_z(\underline{\tilde{z}}) = P_h(R(\underline{\psi}_r, \tilde{z})) \cdot \left|\frac{\partial \tilde{h}}{\partial \tilde{z}}\right| \tag{4.62}$$

Now, in order to derive the probability distribution of the streamflow measurements, we specify the probability distribution of the stage measurements and the equation relating stage and streamflow measurements. We shall first address the selection of the rating curve model and then the choice of stage measurement probability law.

In the interest of keeping the identification problem as simple as possible, we require a simple, parsimonious model of the stage discharge relationship. Linsley, Kohler, and Paulhus (1975) remark that a parabolic relationship is often assumed to be satisfactory, i.e., the relationship is of the form:

$$\tilde{z}_t = b_1(\tilde{h}_t - b_2)^{b_3} \tag{4.63}$$

where h_t and z_t are the stage and streamflow discharge, respectively (at time t) and b_1, b_2, and b_3 are parameters of the function. This function can be inverted to find the R function (Equation 4.59) to be:

$$\tilde{h}_t = \alpha(z_t)^{\gamma} + \beta \tag{4.64}$$

where $\alpha = {}^{1}/_{b_1}$, $\beta = {}^{1}/_{b_2}$, and $\gamma = {}^{1}/_{b_3}$ are the new parameters of interest. Using Equation (4.64), we can derive the Jacobian of the transformation to be:

$$\{J(\underline{\tilde{h}};\underline{\tilde{z}})\}_{ij} = \begin{cases} \alpha\gamma(\tilde{z}_i)^{\gamma-1}, & i = j,\ i = 1, \ldots n \\ \text{zero}, & i \neq j \end{cases} \tag{4.65}$$

and hence:

$$|J(\underline{\tilde{h}};\underline{\tilde{z}})| = (\alpha\gamma)^n \prod_{t=1}^{n} (\tilde{z}_t)^{\gamma-1} \tag{4.66}$$

In order to complete the specification of the probability distribution of the streamflow measurements, the distribution function $P_h(\tilde{h})$ of the stage measurement is required. Since this is unknown, a suitable model for it must be selected. All that we know about the stage error is that it may be either positive or negative and has its most likely value close to zero. Our selected distribution for $\tilde{h}_t$ should reflect the knowledge (or lack of it) that we have about the errors. The distributions that presuppose the minimum information about the error structure for various kinds of knowledge as the error ranges are listed in Table 4.3, which is adapted from Eykhoff (1982). These are also known as the maximum entropy distribution choices. The table indicates that if the error tends to be zero, but the possible range of values is $(-\infty, +\infty)$, the distribution that best reflects our lack of knowledge of the error distributional form is the Gaussian distribution. For practical purposes, since the error in measurement of stage is likely to be small compared to the actual value of the stage itself, we can assume that the errors have unbounded range and hence that the stage measurement

TABLE 4.3

Minimum Apriori Information Choices for
Error Probability Distribution Functions

Error Domain	Probability Distribution
(a,b)	Uniform
$(0,\infty)$	Negative Exponential
(∞,∞)	Gaussian ($E\{e^2\} < \infty$)

probability law can be modeled by a Gaussian distribution. This has the advantage also of providing some degree of mathematical tractability to the problem.

Hence, we shall model the stage measurement probability law as:

$$P_h(\tilde{\underline{h}}) = (2\pi)^{-n/2} \cdot |\Sigma_h|^{-1/2} \cdot \exp\{ \frac{1}{2} \ (\tilde{\underline{h}}-\underline{h})^T \Sigma_h^{-1} \ (\tilde{\underline{h}}-\underline{h})\} \tag{4.67}$$

where $\underline{h}$ is the expected value of $\tilde{\underline{h}}$, (also equal to the true value of $\tilde{\underline{h}}$) and Σ_h is the covariance matrix of $\tilde{\underline{h}}$. Substituting Equations (4.64), (4.66), and (4.67) into Equation (4.62), we obtain:

$$P_z(\tilde{\underline{z}}) = (2\pi)^{-n/2}|\Sigma_h|^{-\frac{1}{2}} \exp \{- \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha^2[(\tilde{z}_i)^\gamma - (z_i)^\gamma] \cdot \{\Sigma_h^{-1}\}_{ij} \cdot$$

$$[(\tilde{z}_j)^\gamma - (zj)^\gamma]\} \cdot (\alpha\gamma)^n \prod_{t=1}^{n} (\tilde{z}_t)^{(\gamma-1)} \tag{4.68}$$

where $\{\Sigma_n^{-1}\}_{ij}$ is the i-jth element of Σ_n^{-1}. All that remains is to specify the covariance matrix Σ_h of the stage measurement.

Now, there is no reason to believe that the stage errors are uncorrelated and homoscedastic. However, the problem of dealing with heteroscedastic errors for Gaussian random variables is not suitably resolved. Hence, in this report, it will be assumed that the stage errors have unknown homogeneous variance. We shall permit, however, the errors to be correlated. Note that even though the stage measurement is homoscedastic, the streamflow measurement will still be heteroscedastic as discussed in Section 4.7.1, due to the nature of the rating curve. Hence, this model

incorporates the case where the streamflow measurements are simultaneously heteroscedastic and correlated. Therefore, the error model is:

$$\tilde{h}_t = h_t + w_t \tag{4.69}$$

and

$$w_t = \sum_{j=1}^{\gamma} \phi_j w_{t-j} + a_t \tag{4.70}$$

where a_t is Gaussian with mean zero and covariance given by:

$$\Sigma\{a_t a_{t+s}\} = \begin{cases} \sigma^2, & s = 0, \; t = 1, \ldots n \\ \text{zero}, & s \neq 0 \end{cases} \tag{4.71}$$

and w_t is the error in the stage measurement. Now Equation (4.70) can be rewritten as:

$$\underline{a} = A \cdot \underline{w} \tag{4.72}$$

where

$$A = \begin{bmatrix} 1 & 0 & \cdots & & 0 \\ -\phi_1 & 1 & & & \vdots \\ -\phi_q & & -\phi_1 & 1 & \vdots \\ 0 & & & & 0 \\ 0 & 0 & -\phi_q & -\phi_1 & 1 \end{bmatrix} \tag{4.73}$$

Hence, we can derive:

$$\Sigma_a = E\{\underline{a}\underline{a}^T\} = A \cdot E\{\underline{w}\underline{w}^T\} \cdot A^T = A\ \Sigma_h\ A^T \tag{4.74}$$

Therefore:

$$\Sigma_a^{-1} = (A^T)^{-1}\ \Sigma_h^{-1}\ A^{-1} \tag{4.75}$$

and hence:

$$\Sigma_h^{-1} = A^T\ \Sigma_h^{-1}\ A \tag{4.76}$$

Substituting Equation (4.76) into Equation (4.67), we get:

$$P\ (\tilde{h}) = (2\pi)^{-n/2}|A^T\ \Sigma_a^{-1}A|^{\frac{1}{2}}\ \exp\{-\tfrac{1}{2}\ \underline{w}^T A^T\ \Sigma_a^{-1}\ A\underline{w}\} \tag{4.77}$$

and using Equation (4.72) and the fact that $|A| = 1$,

$$P_h(\tilde{h}) = (2\pi)^{-n/2}(\sigma^2)^{-n/2}\ \exp\ \{-\tfrac{1}{2}\ \underline{a}^T\ \Sigma_a^{-1}\ \underline{a}\} \tag{4.78}$$

This can be rewritten, using Equations (4.70) and (4.71), as:

$$P_h(\tilde{h}) = (2\pi)^{-n/2}(\sigma^2)^{-n/2}\ \exp\ \{-\frac{1}{2\sigma^2}\ \sum_{t=1}^{n}\ (w_t - \sum_{j=1}^{\gamma}\ \phi_j w_{t-j})^2\} \tag{4.79}$$

Hence, we can derive the probability distribution of the streamflow measurements to be:

$$P_z(\tilde{z}) = (2\pi)^{-n/2}(\sigma^2)^{-n/2}\ \exp\ \{-\frac{1}{2\sigma^2}\ \sum_{t=1}^{n}\ a_t^2\} \cdot (\alpha\gamma)^n\ \sum_{t=1}^{n}\ (\tilde{z}_t)^{\gamma-1} \tag{4.80}$$

where

$$a_t = w_t - \sum_{j=1}^{\gamma} \phi_j \, w_{t-j}$$

and

$$w_t = \alpha[(\tilde{z}_t)^{\gamma} - (z_t)^{\gamma}]$$

Taking the natural logarithm of Equation (4.80) and substituting $z_t(\underline{\theta})$ for z_t, we obtain the log likelihood function (see Section 4.5):

$$L(\theta,\phi,\alpha,\gamma,\sigma,^2 \, |\underline{\tilde{z}},\underline{u}) = -\frac{n}{2} \operatorname{Ln}(2\pi) - \frac{n}{2} \operatorname{Ln}\sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^{n} a_t(\underline{\theta})^{2} + n \operatorname{Ln} \alpha$$

$$+ n \operatorname{\ell n}\gamma + (\gamma-1) \sum_{t=1}^{n} \operatorname{\ell n}(\tilde{z}_t) \qquad (4.81)$$

with a_t as defined in Equation (4.80).

This expression can be simplified by defining:

$$v_t(\underline{\theta},\gamma) = \frac{w_t}{\alpha\gamma} = \frac{(\tilde{z}_t)^{\gamma} - (z_t(\theta))^{\gamma}}{\gamma} \qquad (4.82)$$

and

$$\varepsilon_t(\underline{\theta},\phi,\gamma) = \frac{a_t}{\alpha\gamma} = (v_t - \sum_{j=1}^{q} \phi_j v_{t-j}) \qquad (4.83)$$

Substituting into Equation (4.81) and rearranging:

$$L(\cdot|\tilde{\underline{z}},\underline{u}) = -\frac{n}{2}\,Ln(2\,) - \frac{n}{2}\,Ln\,(\frac{\sigma}{\alpha})^2 + n\,Ln\gamma + (\gamma-1)\sum_{t=1}^{n} \ell n(\tilde{z}_t)$$

$$-\frac{1}{2}\,(\frac{\sigma}{\alpha})^2 \cdot \gamma^2 \sum_{t=1}^{n} \varepsilon_t^{\,2}\,(\underline{\theta},\underline{\Phi},\gamma) \qquad (4.84)$$

In Equation (4.84), we find that parameters α and σ appear only as a ratio and are therefore not uniquely identifiable (they both primarily affect the scaling of the streamflow error). Hence, one of them can be chosen arbitrarily without any effect on the identification. The simplest form of the log likelihood occurs for the choice $\alpha = 1/\gamma$. Substituting this in Equation (4.84), we have the final version of the log-likelihood function:

$$L(\underline{\theta},\Phi,\gamma,\sigma^2|\tilde{\underline{z}},\underline{u}) = -\frac{n}{2}\,Ln(2\pi) - \frac{n}{2}\,Ln\sigma^2 - (1-\gamma)\sum_{t=1}^{n} \ell n(\tilde{z}_t)$$

$$-\frac{1}{2\sigma^2}\sum_{t=1}^{n} \varepsilon_t^2\,(\underline{\theta},\Phi,\gamma) \qquad (4.85)$$

4.7.3 Implementation of the Likelihood Function for Identification

In Equation (4.85), to evaluate ε_t we require v_{t-j} for $j = 1, \ldots q$. In order to compute these terms, we would need to know $\tilde{z}_0, \ldots \tilde{z}_{1-q}$. Since our measurements begin only at time $t = 1$, we must modify the likelihood function so that the summations range only from $t = q + 1$ to $t = n$. Hence, to identify the parameters of the process and probability law models, we must solve the following optimization problem:

$$\max_{\substack{\theta\in\Theta \\ \phi\in\Phi_2 \\ \gamma,\sigma^2>0}} L(\underline{\theta},\phi,\gamma,\sigma^2|D) = -\left(\frac{n-q}{2}\right)Ln(2\pi) - \left(\frac{n-q}{2}\right)Ln\sigma^2 - (1-\gamma)\sum_{t=1}^{n} \ell n(\tilde{z}_t) - \frac{1}{2\sigma^2}\sum_{t=q+1} \varepsilon_t^2(\underline{\theta},\phi,\gamma) \tag{4.86}$$

where

$$\varepsilon_t(\underline{\theta},\phi,\gamma) = V_t(\theta,\gamma) - \sum_{j=1}^{q} \phi_j V_{t-j}(\underline{\theta},\gamma)$$

$$V_t(\underline{\theta},\gamma) = \frac{(\tilde{z}_t)^\gamma - (\tilde{z}_t(\underline{\theta}))^\gamma}{\gamma}$$

$$z_t(\underline{\theta}) = M(\underline{\theta},u(t))$$

In order to implement this function, the order q of the autoregressive stage measurement error model must be selected in advance. It is unlikely that a model of order greater than 2 or 3 will ever be necessary. It is recommended that at first, order q = 1 should be selected and that a higher order be used only if an analysis of the residuals ε_t^* of the identified model reveals significant autocorrelation.

4.8 Experimental Testing

4.8.1 Introduction

In Section 4.7, a new likelihood procedure for the identification of watershed model parameters was presented. The new procedure is based on the principle that, in many cases, the streamflow values are not directly measured but are obtained from measured stage (river depth) readings through a nonlinear rating curve transformation. If the errors in the measured stage

values are assumed to be approximately Gaussian, then, due to the nonlinear nature of the rating curve transformation, the errors in the recorded streamflow values will be positively biased (skewed error distribution) and have nonstationary variances. If a measure of closeness such as the simple least squares (SLS) function which treats all errors as having constant variance were to be used with such data, we would expect the parameters of the model to be biased. On the other hand, if the correct likelihood criterion for such measurement data were to be used, we would expect the parameter estimates to be close to their true values. In the studies reported here, the applicability of the new likelihood procedure to the identification of the parameters of the model SIXPAR was examined, and its behavior compared to that of the SLS function. The case studied was where the stage measurement errors were Gaussian, uncorrelated, and homoscedastic. Hence, only one extra parameter, the rating curve transformation parameter γ, must be estimated from the data. The optimization procedure used was the Levenberg-Marquardt procedure (Goldstein and Larimore, 1980). The negative of the likelihood function was minimized.

4.8.2 Identifiability of the Transformation Parameter γ

First, the ability to precisely identify the correct value of the rating curve transformation parameter γ was analyzed. The parameters UM, UK, BM, BK, A, and X of model SIXPAR were assumed to be known and were set equal to their true values.

The first study was conducted with true value of $\gamma = 1.0$. This corresponds to the case where the rating curve is linear and hence the streamflow errors are Gaussian, independent, and homoscedastic. An input data set RAIN-1 of sample length $n = 50$ (see Appendix 7) was used to generate the

observed streamflow sequences corrupted with measurement error according to the procedure described in Appendix 8. Ten different realizations of measured streamflow sequences were used in the study. The variance of the stage measurement error was chosen to be 0.25. For each realization of the observed streamflow sequences, the maximum likelihood estimate γ^* of the transformation parameter γ was obtained through optimization of the likelihood function starting at the true value $\gamma = 1.0$. The results are tabulated in Table 4.4. Notice that the values of γ^* are distributed between 0.59 and 1.61 (a width of 0.57) with a mean value of 0.94. This is a rather wide variation and suggests poor precision of the parameter γ. For eight out of the ten results, the theoretical approximate 95% confidence interval includes the true value of $\gamma = 1.0$, indicating that the wide variation in the results is due essentially to low sensitivity of the likelihood function to variations in the parameter γ.

A similar study to the one described above was conducted, this time with the true value of $\gamma = 0.5$, indicating a nonlinear rating curve. The results are presented in Table 4.5. The range of values of γ^* is between 0.15 and 0.67 (a width of 0.52) with a mean value of 0.44. The precision of the parameter estimates of γ is of the same order as in the previous study.

In order to determine the reason for the poor sensitivity of the parameter γ, the concentration of flow-stage values along the rating curve was plotted for $\gamma = 0.5$ (see Figure 4.4). The figure indicates that the data values are clustered closely together at the lower end of the rating curve with few values towards the upper end. Clearly, the data set does not contain much information about the nature of the transformation. A similar plot (see Figure 4.5) was created using a new precipitation data set RAIN-4 of sample length 50 which has different characteristics (see Appendix 4).

TABLE 4.4

Maximum Likelihood Estimates for the Transformation Parameter γ. (True value = 1.0; data set used = RAIN-1)

Realization	γ^*	$\pm 2\sigma$ Interval	Function	Iterations
1	1.1293	±0.18	38.539	7
2	0.800	±0.17	37.516	9
3	0.9768	±0.18	29.244	8
4	0.5931	±0.19	34.719	18
5	0.9209	±0.18	37.073	8
6	0.9813	±0.17	26.800	5
7	0.9049	±0.19	35.755	10
8	1.1595	±0.18	28.563	5
9	1.0098	±0.19	21.074	6
10	0.8939	±0.17	31.411	8
Average	0.9370	±0.18	32.069	8.4

TABLE 4.5

Maximum Likelihood Estimates for the Transformation Parameter γ. (True value = 0.5; data set used = RAIN-1)

Realization	γ^*	$\pm 2\sigma$ Interval	Function	Iterations
1	0.6264	±0.17	73.133	7
2	0.3059	±0.17	71.447	9
3	0.4609	±0.18	64.540	9
4	0.1470	±0.18	70.066	15
5	0.4296	±0.18	70.849	8
6	0.4801	±0.17	62.160	6
7	0.3733	±0.19	69.766	11
8	0.6720	±0.17	62.311	5
9	0.4945	±0.19	55.007	5
10	0.3966	±0.17	65.640	5
Average	0.4386	±0.18	66.492	8.3

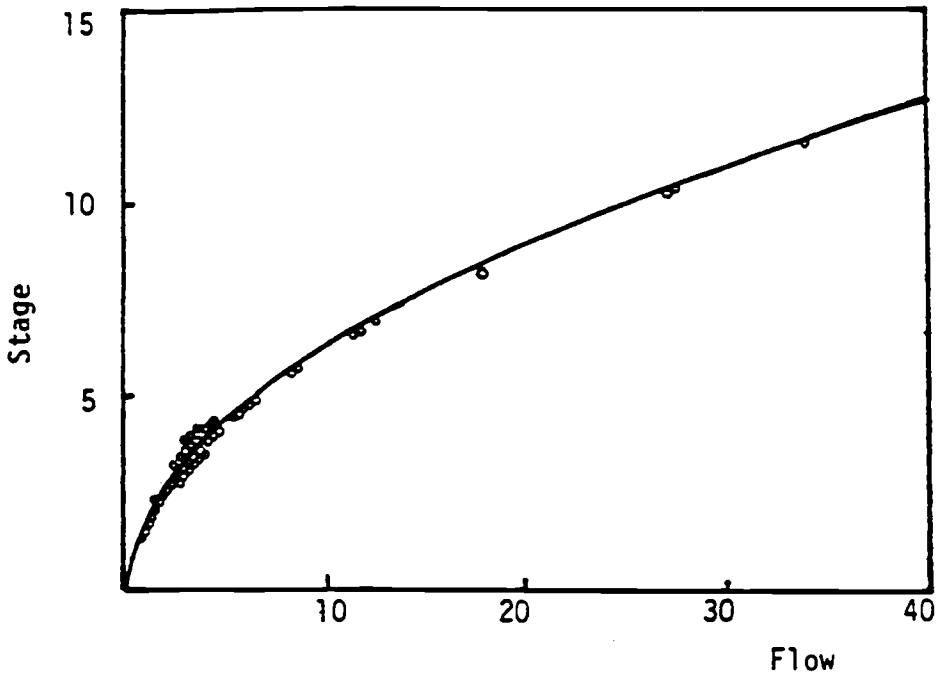


Figure 4.4. Distribution of data points of flow (stage) for data set RAIN-1; $\gamma = 0.5$

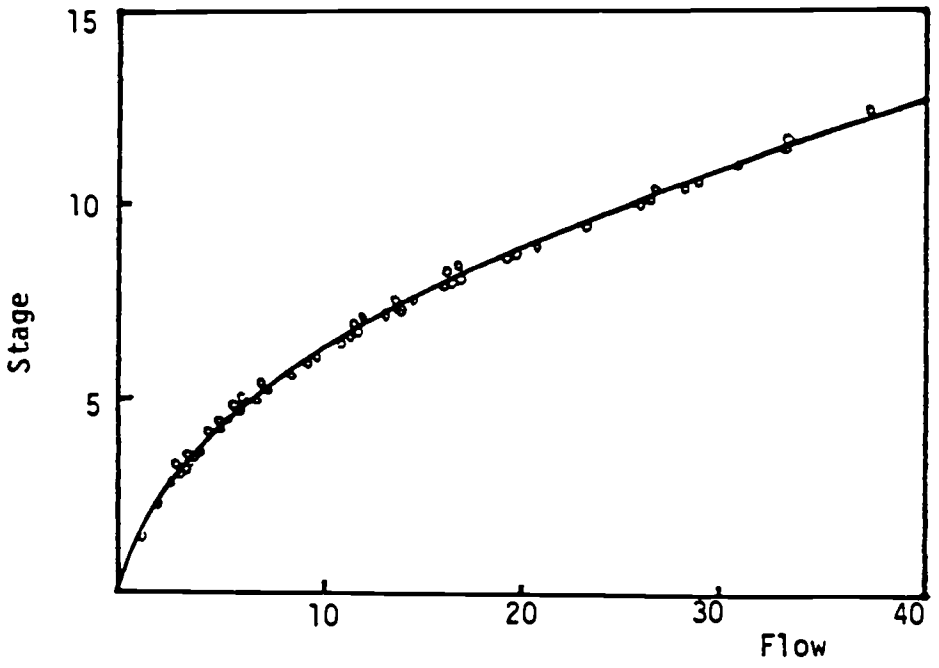


Figure 4.5. Distribution of data points of flow (stage) for data set RAIN-4; $\gamma = 0.5$

The figure indicates that the new data set should contain information that is more likely to result in precise estimates for the parameter γ.

A third study was therefore conducted using the new data set with γ = 0.5. The results are presented in Table 4.6. The 95% confidence interval has reduced significantly in width from approximately ±0.17 to approximately ±0.11.

The results of these studies clearly emphasize that, as with any parameter estimation problem, some care must be taken to ensure that the chosen data set contains sufficient information to ensure the identifiability of the parameters of interest. In this case, the identifiability of the rating curve transformation parameter is sensitive to the extent of hydrologic variability in the observed data.

Finally, the shape of the negative likelihood function in the parameter space is plotted in Figure 4.6. The function appears to be well-behaved, convex, and unimodal with respect to parameter γ, although a little insensitive to parameter variation in the region of the optimum.

4.8.3 Sensitivity of the Model Parameters to the Value of γ

In this section, the sensitivities of the optimal values of each of the model parameters to variations in the value of γ from its true value were studied. The objective was to discover which of the model parameters were most sensitive to the properties of the random errors in the data. Unfortunately, financial limitations precluded a statistical approach to this analysis. Hence, the results presented here each use only a single realization of observed streamflow values.

The first study was conducted using the new data set RAIN-4 and error realization #6 (Table 4.6) for which the maximum likelihood estimate of

TABLE 4.6

Maximum Likelihood Estimates for the Transformation Parameter γ. (True value = 0.5; data set used = RAIN-4)

Realization	γ^*	$\pm 2\sigma$ Interval	Function	Iterations
1	0.4744	±0.11	96.64	11
2	0.3757	±0.12	95.71	12
3	0.6419	±0.12	85.77	17
4	0.5696	±0.11	95.20	13
5	0.4391	±0.12	93.53	17
6	0.4961	±0.11	84.80	6
7	0.4913	±0.12	92.92	8
8	0.5762	±0.11	86.46	14
9	0.4187	±0.11	77.58	15
10	0.1526	±0.12	85.63	27

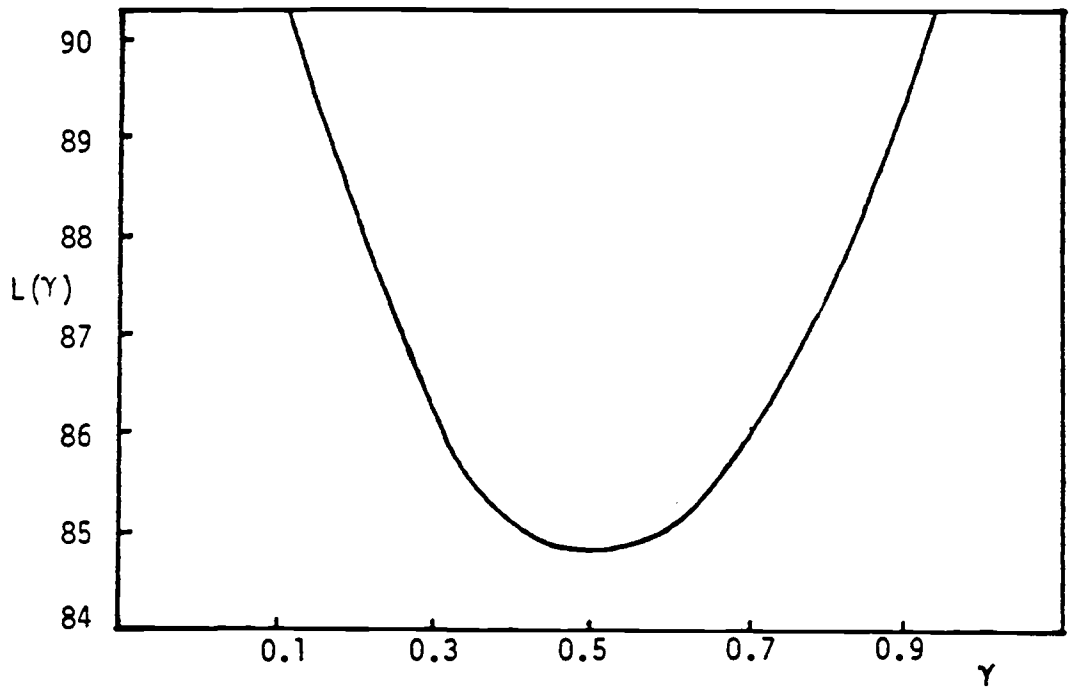


Figure 4.6. Plot of the likelihood function as γ is varied for Realization #6 of Table 4.6

γ was closest to its true value of $\gamma = 0.5$. The value of γ was varied from -2 to +2 and for each value (holding γ fixed), the likelihood function was used to estimate the optimal values of each of the parameters UM, UK, BM, and BK individually, starting from their true values (three of the four parameters are held fixed at their true values, while a one-dimensional search was conducted for the other). The results are plotted in Figure 4.7. The circles joined by solid links represent the optimal values of each parameter as γ is varied, while the dashed lines represent their approximate two-standard deviation confidence intervals (whenever they fit inside the boxes). The results indicate that the parameters are more sensitive to $\gamma < 0$ than they are to $\gamma > 0$. For $\gamma > 0$, only the parameter BM appears to be sensitive to the value of γ, while parameters UM, UK, and BK are only slightly affected.

In the previous study, the confidence intervals on the parameters are rather wide. It was decided to see if improving the parameter precision would lead to greater sensitivity of the parameters to variations in γ. The data set used was of sample length $n = 50$. A third data set, RAIN-5, was created by repeating the sequence of values in data set RAIN-4 ten times in succession. The new data set is therefore of sample length $n = 500$. Figure 4.8 contains the results of the analysis conducted using the data set RAIN-5 for true value of $\gamma = 0.5$. The precision of the parameter estimates has clearly improved. However, the parameter values do not seem to be very sensitive to variations of γ in the range zero to 1 (Note: $\gamma = 0$ corresponds to a logarithmic relationship between the flows and stage values, while $\gamma = 1$ corresponds to a linear relationship and is equivalent to use of the SLS function).

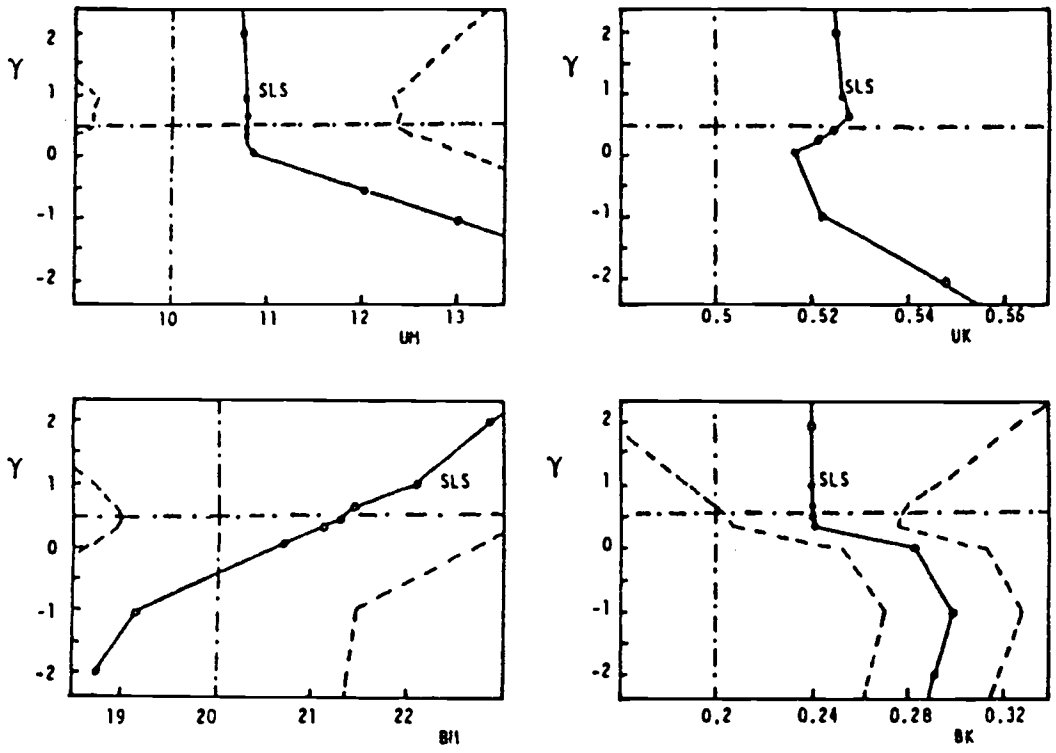


Figure 4.7. Sensitivity of the parameters to perturbations of γ for data set RAIN-4 (n = 50)

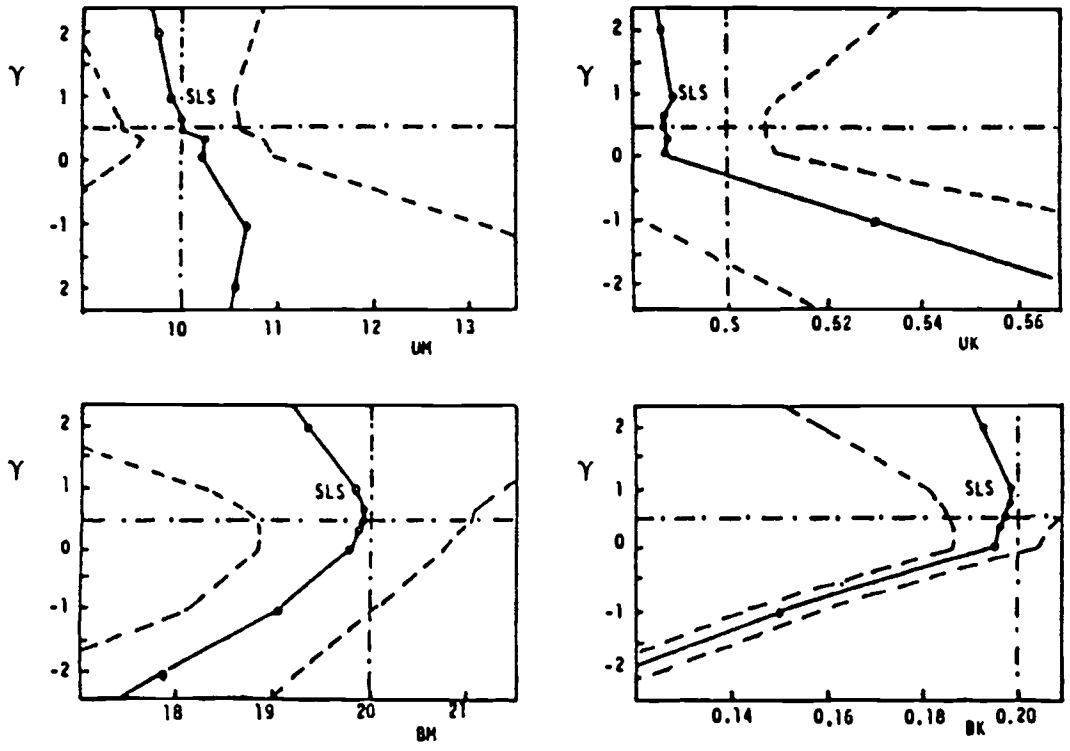


Figure 4.8. Sensitivity of the parameters to perturbations of γ for data set RAIN-5 (γ = .5)

Finally, an identical analysis to the one above was conducted but with true value of $\gamma = 0.05$ (virtually a logarithmic transformation). This study aimed to see if the parameters would be more sensitive to variations in γ if the nonlinearity in the rating curve was higher ($\gamma << 1$). The results are presented in Figure 4.9. The sensitivities of the parameters to variations in γ are significantly higher. Once again, the most noticable variation between $\gamma = 1.0$ and $\gamma = 0.05$ (true value) is in the parameter BM.

The analysis conducted here has no statistical basis and the results are therefore not conclusive. However, the results obtained seem to suggest that, unless the data set is long enough and information and the rating curve are significantly nonlinear, the likelihood function approach may not give significantly different parameter estimates from those obtained using the SLS function.

4.8.4 Joint Identification of the Model and Transformation Parameters

In this section, the performance of the likelihood function was compared to that of the SLS function with regards to their abilities to jointly identify the four parameters UM, UK, BM, and BK of the model SIXPAR. The true parameter values used were UM = 10.0, UK = 0.5, BM = 20.0, BK = 0.2, A = 0.3107, X = 3.0. Parameters A and X were fixed at their true values. The input data set RAIN-5 of sample length n = 500 was used, and the true value of the rating curve transformation parameter was chosen to be $\gamma = 0.05$.

In the first study, ten realizations of observed streamflow sequences were generated using the procedure outlined in Appendix 8. Using each of these data sets, each of the two functions, the likelihood function and the SLS function, were used to optimize the parameter values. The results

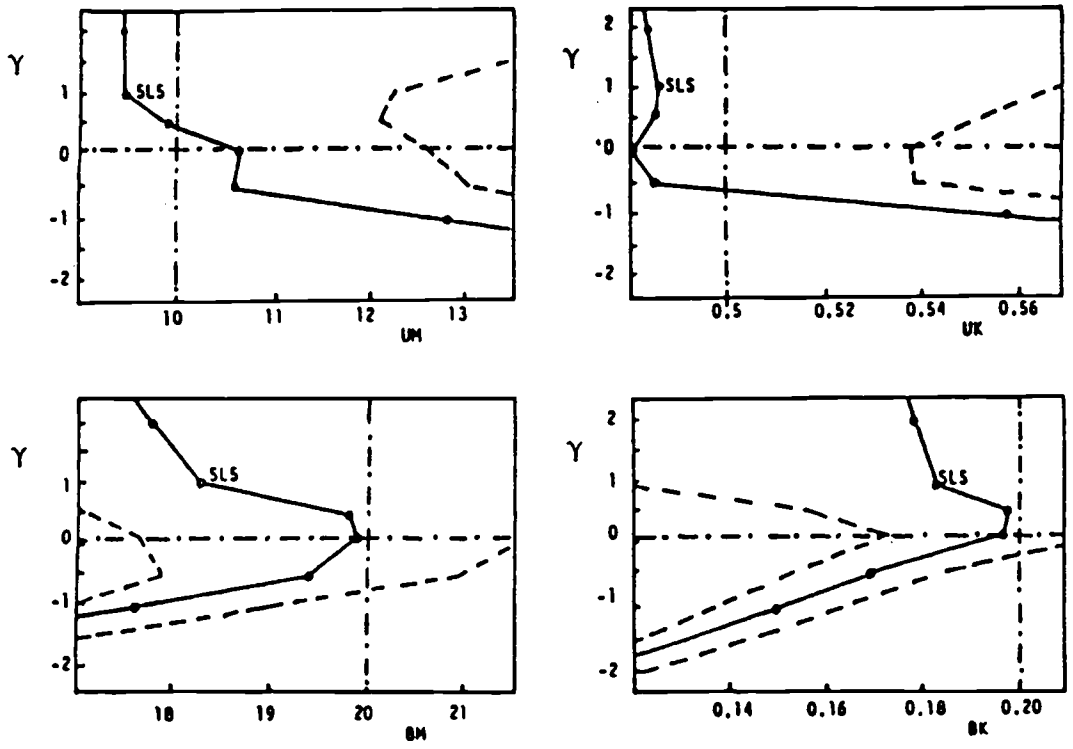


Figure 4.9. Sensitivity of the parameters to perturbations of γ for data set RAIN-5 ($\gamma = .05$)

should therefore indicate the location of the global optimum parameter sets for each function. The results of the likelihood function optimization runs are presented in Table 4.7, while the SLS results are presented in Table 4.8. The results are quite interesting. First, the large number of iterations required for the optimization techniques to converge indicate that the surface is not well approximated by a quadratic function, even so close to the optimum. In the case of the likelihood function, this could understandably be due to the nonlinear rating curve transformation. The reason for this behavior in the case of the SLS function is unclear. Second, in keeping with the results of the previous subsection, only the estimate of parameter BM seems to be significantly affected when the SLS function, which is not the correct (maximum likelihood) measure of closeness for this error case, is used. In general, parameters UM and UK have been better estimated by the SLS function, while BM and BK have been better estimated by the likelihood function. The estimate for γ seems to be generally good.

In order to establish which measure of closeness is identifying a model closer in behavior to the true model, the sum of the squares of the errors (SSE) between the model output using each estimated parameter set and the true flows was computed. The results appear in Table 4.9. In six out of ten cases, the likelihood criterion performed better, while in four out of ten cases, the SLS function performed better. With only a sample of ten results, we cannot conclude that the likelihood function is superior on the basis of this small amount of discrimination. However, the fact that the SLS poorly estimates parameter BM almost every time indicates that the likelihood function is probably more reliable.

In the previous analysis, the SLS function had to estimate only four parameters, while the likelihood function had to estimate five. In order to

TABLE 4.7

Optimal Parameter Values Using the Likelihood Function
(Search was initialized from the true values).

Realization	UM	UK	BM	BK	γ
1	13.9748	0.4033	18.2656	0.1839	0.0352
2	12.1503	0.4476	19.8877	0.1713	0.0208
3	11.0342	0.4739	20.9160	0.1941	0.0911
4	14.5609	0.4452	23.6113	0.1612	0.0360
5	7.9599	0.5230	20.630	0.1760	0.0589
6	10.9164	0.4820	17.7351	0.2103	0.0957
7	14.8503	0.4311	15.8732	0.2359	0.0659
8	12.0975	0.4689	20.0041	0.1630	0.0780
9	9.8622	0.5173	21.8333	0.1952	0.0002
10	9.9698	0.4676	14.6358	0.2563	0.0180
Average	11.7376	0.4660	19.3393	0.1947	0.0500

TABLE 4.8

Optimal Parameter Values Using the SLS Function
(Search was initialized at the true values).

Realization	UM	UK	BM	BK
1	13.9970	0.4161	10.8460	0.2912
2	9.7493	0.5078	20.1585	0.1788
3	11.7275	0.3999	9.4106	0.3679
4	14.5743	0.4026	12.4574	0.2593
5	7.6813	0.5795	15.8067	0.2160
6	9.4910	0.5602	21.1586	0.1993
7	11.0904	0.5066	15.6357	0.2447
8	10.8901	0.4956	14.2535	0.2299
9	10.1724	0.5705	22.0250	0.1951
10	10.7344	0.4661	13.4312	0.2494
Average	11.0108	0.4905	15.5183	0.2431

TABLE 4.9

Sum of Squares of Differences Between Model Output and True Flows for the Parameters of Tables 4.7 and 4.8

Realization	SSE	
	Likelihood	SLS
1	60.5509	42.9708
2	20.4641	18.4183
3	17.3307	108.5244
4	206.4932	53.0738
5	127.7529	181.2929
6	5.1191	49.9836
7	149.5690	47.9628
8	15.8979	40.4056
9	18.6687	120.7812
10	53.1610	56.0662
Average	67.5008	71.9480

ensure that the likelihood function performance was not hampered by having to estimate an extra parameter, the likelihood runs were repeated with the transformation parameter γ fixed at its true value. The results, presented in Table 4.10, are not significantly different from those when γ was also assumed unknown. The sums of squares of errors (SSE) between model and true flows are also similar, except that run #2 now ranks better than the SLS function using this measure. Clearly, parameter interaction between γ and the other model parameters is not a significant factor influencing the results that were obtained.

Finally, using the same data sets, the two functions (likelihood and SLS) were used to optimize the parameters using initial values of UM through BK obtained by a -20% perturbation from their true values. The value of γ was initialed at the value $\gamma = 1.0$ to reflect lack of prior knowledge regarding the nature of the rating curve transformation (Note: with $\gamma = 1.0$, the likelihood and SLS functions are equivalent). The parameter estimates are presented in Tables 4.11 and 4.12, while the SSE values appear in Table 4.13. Only six error realizations were used in this study due to financial restrictions on available computer time. In nearly every case, the algorithm termination point is different from that obtained when the optimization began at the true values. Again, the likelihood function has generally performed better in identifying the parameter BM. With respect to the SSE measure, however, the two functions come out even, each being superior in exactly three of the cases.

4.8.5 Conclusions and Comments

The results of the studies in this subsection have demonstrated that the likelihood function proposed is a viable alternative to the SLS

TABLE 4.10

Optimal Parameter Values Using the Likelihood Function (Fixed = 0.05; search from true parts).

Realization	UM	UK	BM	BK	SSE
1	14.0815	0.4020	18.0099	0.1848	61.0795
2	12.0571	0.4513	20.2530	0.1692	16.2592
3	11.0548	0.4751	21.0014	0.1925	17.4721
4	14.5198	0.4389	23.6286	0.1622	199.9001
5	7.9552	0.5230	20.6437	0.1759	5.7604
6	10.8136	0.4870	17.5692	0.2124	5.7604
7	15.5305	0.4193	15.5744	0.2410	186.7448
8	12.0986	0.4693	19.9806	0.1630	16.0762
9	9.9036	0.5150	21.7702	0.1961	18.4141
10	10.1345	0.4847	15.4281	0.2441	24.6646
Average	11.8149	0.4666	19.3859	0.1941	67.4891

TABLE 4.11

Optimization Runs Using the Likelihood Function Started at the -20% Perturbation Parameter Values (γ started at value = 1.0)

Realization	UM	UK	BM	BK	γ
1	19.1083	0.3433	12.4391	0.1837	0.3474
2	11.8822	0.3978	16.8945	0.1781	0.0160
3	12.0956	0.4088	15.6914	0.2341	0.1068
4	14.7917	0.4388	23.5410	0.1624	0.0441
5	9.8163	0.3827	17.0132	0.1538	0.0541
6	12.9820	0.4049	18.3245	0.1542	0.1441

TABLE 4.12

Optimization Runs Using the SLS Function Started at the -20% Perturbation Parameter Values

Realization	UM	UK	BM	BK
1	17.2139	0.3824	10.1802	0.2065
2	11.1836	0.4140	15.5229	0.1864
3	11.7279	0.3996	9.4531	0.3665
4	13.8583	0.4127	15.2470	0.2203
5	7.4580	0.5863	15.6392	0.2187
6	14.1836	0.3822	17.3525	0.1449

TABLE 4.13

Sum of Squares of Differences Between Model Output and True Flows for the Parameter of Tables 4.11 and 4.12

Realization	SSE	
	Likelihood	SLS
1	137.928	67.054
2	113.920	151.513
3	17.472	108.260
4	220.833	41.162
5	388.348	202.070
6	61.811	82.959

function, provided that sufficient information exists in the data to identify the rating curve transformation parameter γ. However, its superiority to the SLS as an appropriate measure of closeness for watershed models has not been conclusively demonstrated. The main obstacle to achieving this goal was the high cost of the computer runs and the financial restrictions on available computer time. A thorough analysis would require many more runs with different error realizations to establish a statistically significant sample from which to draw conclusions. Also, the results are likely to be sensitive to the magnitude of the error variance used and to the degree of nonlinearity (value of γ_{true}) of the rating curve transformation. Another point that should be considered is that the four parameters identified in this study are all highly sensitive, and are therefore less affected by the characteristics of the data error than less sensitive parameters would be. Further testing of the likelihood function on the model SIXPAR and other watershed models is required.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The objective of this research project was to investigate methods for improving the reliability of compartmental models such as conceptual rainfall-runoff models. The study focused on the structural nonidentifiability problems of such models which have been observed to cause severe parameter estimation difficulties. In particular, the problems arising due to the threshold nature of some parameters have been investigated.

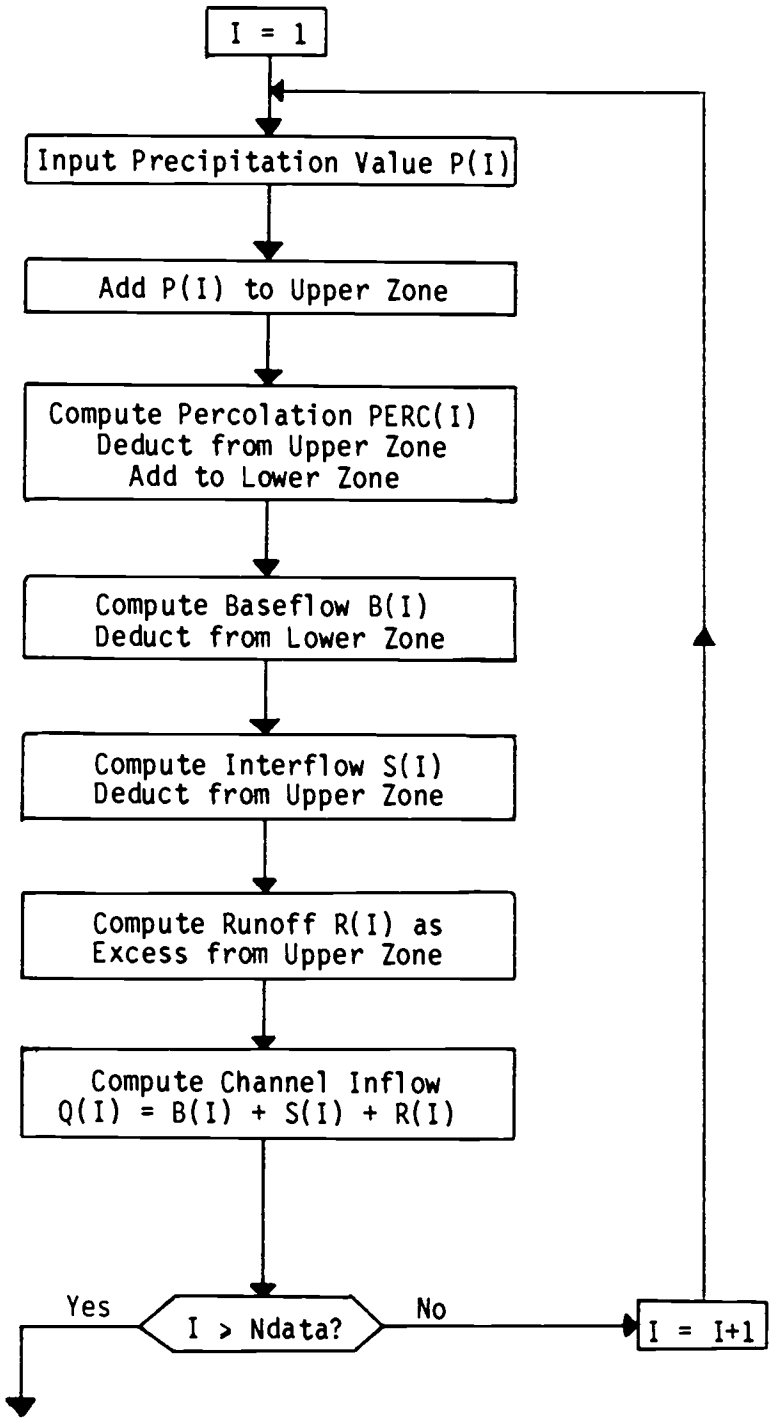
The model chosen as the focus of this study was the Soil Moisture Accounting model of the U.S. National Weather Service. A simplified version of the model was developed in order to simplify the problem and so that we could focus on the important aspects of the problem. The simplified version, called SIXPAR, has all the major features of the NWSRFS model. Studies on the model revealed that the problems with identification of the percolation equation parameters of the model were not due to inadequate data but are inherent in the formulation of the model. Special tools were developed that make it possible to detect such problems without the need for expensive and time-consuming response analyses. These tools were used to demonstrate the superiority of a reparameterized version of the percolation equation.

Three new mathematical procedures have been developed as part of this research project. First, a measure of structural identifiability has been constructed which makes it possible to identify those parameters that are responsible for poor parameter identifiability of a model. This measure permits such studies to be conducted in the multiparameter space which cannot be easily done using the traditional methodology of graphical response

surface analysis. Second, we have shown that derivatives can be explicitly computed for such models, therefore removing the need for methods based on approximations. The mathematics developed were used to investigate the identifiability of the threshold structure of a linear reservoir. The analysis demonstrates theoretically what constitutes "informative" data for such models. Finally, a new objective function for rainfall-runoff models was developed using likelihood theory and based on the error structure arising due to the nonlinear nature of the rating curve. This procedure avoids some of the theoretical problems associated with previous approaches.

APPENDIX 1

OPERATIONAL OUTLINE OF THE MODEL SIXPAR



APPENDIX 2

COMPUTER PROGRAMS

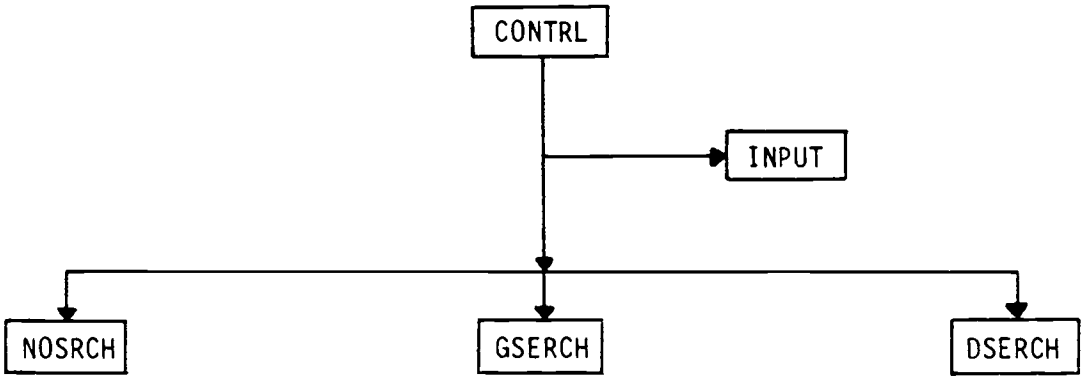
A-2.1 Introduction

The computer programs used in this work were written in the language FORTRAN. The programs were run on a Digital Electronics Corporation DEC-20 system at Case Western Reserve University.

The computer programs are included in Section A-2.3. An explanation of the subroutine call structure of the programs is included in Section A-2.2. There are two sets of programs. The first set includes the programs set up to identify the parameters of the model SIXPAR. The second set of programs is for the search algorithm based on transformations of the parameter space.

Please note that some of the subroutines require double precision.

A-2.2. Subroutine Call Structures



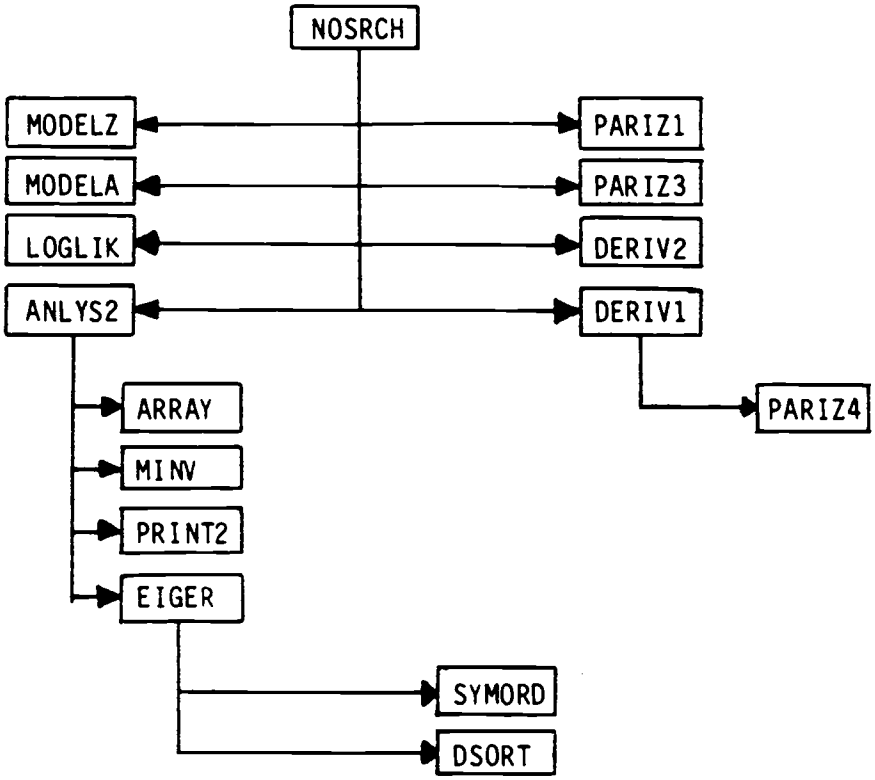
CONTRL - Main calling program; decides on type of run

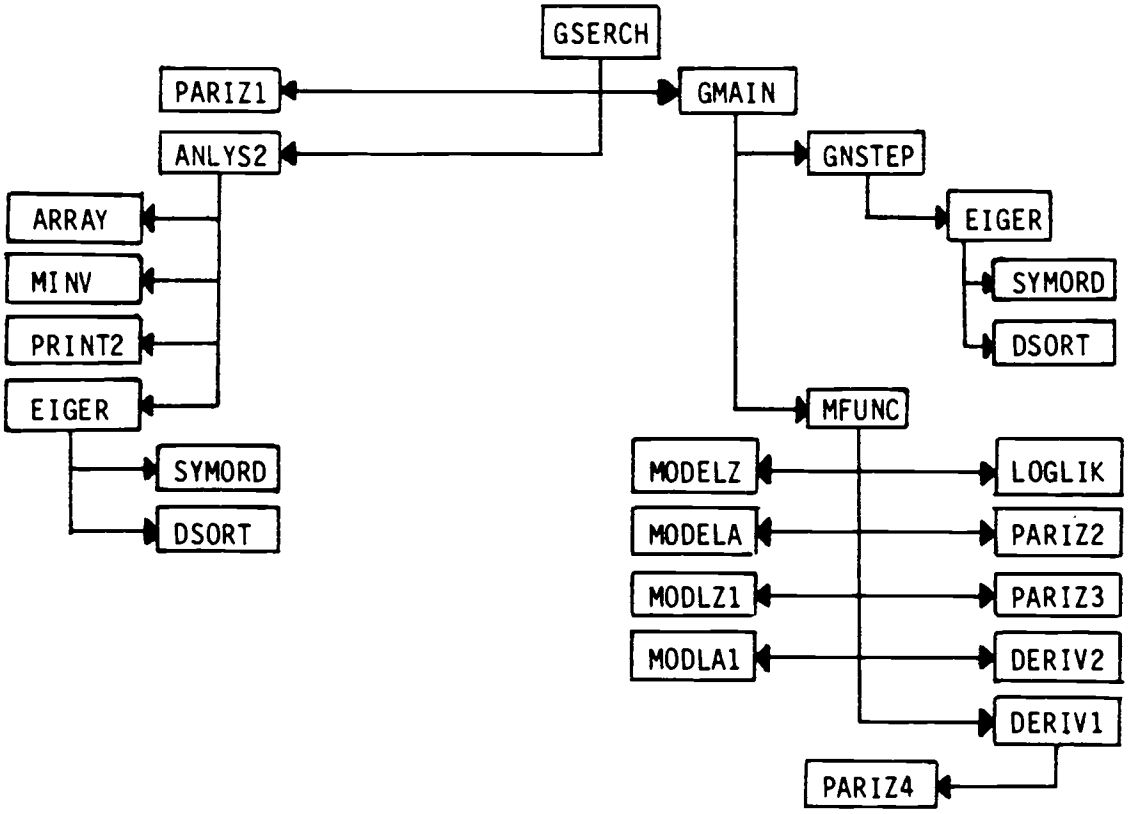
INPUT - Reads Control file and inputs data

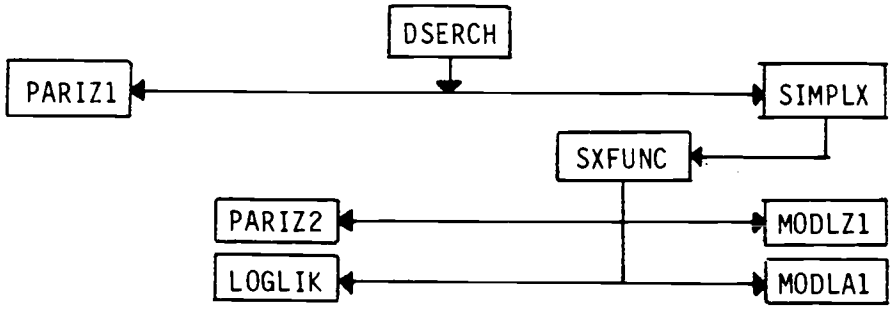
NOSRCH - Controls non-optimization run; conducts the Sensitivity-Identifiability analysis

GSERCH - Controls gradient algorithm optimization run

DSERCH - Controls direct-search algorithm optimization run


NOSRCH
MODELZ
MODELA
LOGLIK
ANLYS2
PARIZ1
PARIZ3
DERIV2
DERIV1
PARIZ4
ARRAY
MINV
PRINT2
EIGER
SYMORD
DSORT


GSERCH
PARIZ1
ANLYS2
GMAIN
GNSTEP
EIGER
SYMORD
DSORT
ARRAY
MINV
PRINT2
EIGER
SYMORD
DSORT
MFUNC
MODELZ
MODELA
MODLZ1
MODLA1
LOGLIK
PARIZ2
PARIZ3
DERIV2
DERIV1
PARIZ4



MODELZ - Original parameterization of model SIXPAR; computes flows and partial derivatives

MODLZ1 - Original parameterization of model SIXPAR; computes only flows

MODELA - Reparameterized model SIXPAR (i.e., reparameterized percolation equation); computes flows and partial derivatives

MODLA1 - Reparameterized model SIXPAR; computes only flows

LOGLIK - Computes value of the chosen measure of closeness (SLS or likelihood function)

GMAIN - Gradient optimization algorithm incorporating Levenberg-Marquardt modifications; based on the Newton approach

GNSTEP - Computes the optimal Levenberg-Marquardt step in the parameter space for GMAIN

MFUNC - Controls computation of the function value and the first and second partial derivatives at the given parameter values

DERIV1 - Computes gradient vector

DERIV2 - Computes information matrix

PARIZ1 - Determines the reparameterization to be used and converts from parameter names to a parameter vector for the optimization algorithms

PARIZ2 - Inverts to the original parameterization

PARIZ3 - Reparameterizes the model derivatives

PARIZ4 - Reparameterizes the model derivatives in the case when the likelihood function based on the rating curve is used

SIMPLX - Controls the operation of the Simplex direct search optimization algorithm of Nelder and Mead

SXFUNC - Controls computation of the function value for SIMPLX

ANLYS2 - Conducts the Sensitivity-Identifiability analysis

ARRAY - Used in inversion of a matrix

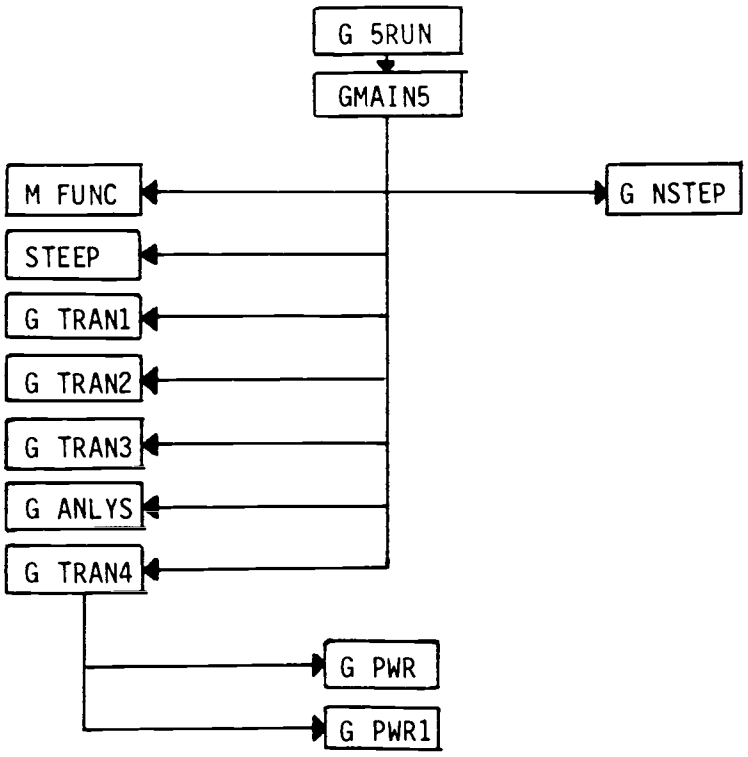
MINV - Used in inversion of a matrix

PRINT2 - Prints out the results of the Sensitivity-Identifiability results

EIGER - Controls computation of eigenvalues and eigenvectors

SYMORD - Computes eigenvalues and eigenvectors using Householder's transformation

DSORT - Sorts eigenvalues and eigenvectors in order of decreasing eigenvalue magnitude



G 5RUN - Main calling program

GMAIN5 - Control operation of algorithm based on transformations

G TRAN1 - Computes transformed parameter values

G TRAN2 - Computes transformed gradients and Hessian

G TRAN3 - Retransforms parameters to original space

G TRAN4 - Computes best transformation parameters

G PWR - Computes derivatives for transformation parameter search

G PWR1 - Computes derivatives for transformation parameter search

ERROR - Computes backward quadratic prediction error

G ANLYS - Same as ANLYS2

STEEP - Conducts a steepest descent search

Main Calling Program

```
00100   C
00200   C  ***  CONTROL PROGRAM FOR THE OPERATION OF THE  ***
00300   C  ***  CONCEPTUAL RAINFALL-RUNOFF MODEL.        ***
00400   C
00500   C------------------------ ------------------------------
00510   C
00520           COMMON/CNTROL/ISERCH, IOP, ISENSE, IMODEL, IREPAR
00530   C
00540   C------------------------ ------------------------------
00600   C
00700   C  ***  OBTAIN CONTROL INFO. FOR THE RUN, AND THE INPUT
00800   C  ***  DATA.
00900   C
01000           CALL INPUT
01100   C
01150   C-------------------------------------------------------
01155   C
01160           IF(ISERCH.EQ.2)GO TO 100
01162           IF(ISERCH.EQ.3)GO TO 110
01165   C
01170   C------------------------ ------------------------------
01200   C
01300   C  ***  NON-OPTIMIZATION RUN. EVALUATION OF STATISTICS AND
01400   C  ***  IDENTIFIABILITY/SENSITIVITY MEASURES AT THE GIVEN
01500   C  ***  PARAMETER SET.
01600   C
01700           CALL NOSRCH
01800   C
01900           GO TO 200
01950   C
02000   C-------------------------------------------------------
02100   C
02101   100     CONTINUE
02102   C
02110   C  ***  OPTIMIZATION RUN, USING THE LEVENBERG - MARQUARDT
02120   C  ***  MODIFICATION OF THE NEWTON TYPE GRADIENT SEARCH
02130   C  ***  ALGORITHM .
02140   C
02150           CALL GSERCH
02160   C
02162           GO TO 200
02164   C
02170   C------------------------ ------------------------------
02171   C
02172   110     CONTINUE
02173   C
02174   C  ***  OPTIMIZATION USING THE SIMPLEX DIRECT SEARCH NONLINEAR
02175   C  ***  MINIMIZATION ALGORITHM.
02176   C
02177           CALL DSERCH
02178   C
02179   C------------------------ ------------------------------
02180   C
02190   200     CONTINUE
02195   C
02200           STOP
02300           END
02400   C
```

```
00100   C zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz
00200   C
00300           SUBROUTINE INPUT
00400   C
00500   C-----------------------------------------------------------------------
00600   C
00700   C  ***  READS CONTROL INFORMATION AND INPUT DATA TO BE USED
00800   C  ***  IN THE VARIOUS OPERATIONS OF THE PROGRAM.  AND PRINTS
00900   C  ***  THE INFORMATION RELEVANT.
01000   C
01100   C-----------------------------------------------------------------------
01200   C
01300           DOUBLE PRECISION JINFIL, JOUTFL, JCTLFL, JOBQFL
01400   C
01500           COMMON/RINPUT/NDATA, P(999), OBSQ(999), TRUQ(999)
01600           COMMON/ROUTPT/Q(999), R(999), S(999), B(999), PPERC(999), PERC(999)
01700           COMMON/STATES/US(999), BS(999)
01800           COMMON/PARAMS/NPAR, UM, UK, BM, BK, Z, X, USI, BSI
01900   C
02000           COMMON/CNTROL/ISERCH, IOP, ISENSE, IMODEL, IREPAR
02050           COMMON/SERCH1/ITMAX, INPOL, PAREPS, FNEPS, STPMAX, PLM, PLMAX
02075        1  , PLMULT, PLDIVD
02100   C
02110           COMMON/PARAM2/GAMA, PHI1, SIGSQR
02120   C
02150   C-----------------------------------------------------------------------
02200   C***       COMMON/GRIDS/IPAR, JPAR, PAR1, PAR2, INITAL, PMAX1, PMAX2
02300   C***    1  , STEP1, STEP2
02400   C***       COMMON/JOIN/IUM, IUK, IBM, IBK, II, IX
02500   C***       COMMON/INFACE/IDEX, IENDEX, PAR(6), PERCNT
02600   C***       COMMON/DESDAT/ALPHA, NUMPTS, NTERMS, SCALE(6)
02700   C***       COMMON/BUGOUT/IBUG
02800   C
02900   C-----------------------------------------------------------------------
03000           WRITE(5,100)
03100   100     FORMAT(' PRECIPITATION INPUT FILE-? ')
03200           READ(5,110)JINFIL
03300           WRITE(5,110)JINFIL
03400   110     FORMAT(A10)
03500           OPEN(UNIT=20,FILE=JINFIL)
03600   C-----------------------------------------------------------------------
03700           WRITE(5,105)
03800   105     FORMAT(' OBSERVED FLOWS FILE-?')
03900           READ(5,106)JOBQFL
04000           WRITE(5,106)JOBQFL
04100   106     FORMAT(A10)
04200           OPEN(UNIT=21,FILE=JOBQFL)
04300   C-----------------------------------------------------------------------
04400   C        WRITE(5,120)
04500   C120     FORMAT(' OUTPUT FILE-?')
04600   C        READ(5,130)JOUTFL
04700   C        WRITE(5,130)JOUTFL
04800   C130     FORMAT(A10)
04900   C        OPEN(UNIT=22,FILE=JOUTFL)
05000   C-----------------------------------------------------------------------
05100           WRITE(5,140)
05200   140     FORMAT(' CONTROL FILE-? ')
05300           READ(5,150)JCTLFL
05400           WRITE(5,150)JCTLFL
05500   150     FORMAT(A10)
05600           OPEN(UNIT=23,FILE=JCTLFL)
05700   C-----------------------------------------------------------------------
05800           WRITE(5,200)
05900   200     FORMAT(' SIMULATION USING 6-PARAMETER MODEL  ')
06000           WRITE(5,210)JINFIL
06100   210     FORMAT(' PPT. INPUT FILE IS --- ',A10)
06110           WRITE(5,212)JOBQFL
06120   212     FORMAT(' OBS. FLOWS FILE IS --- ',A10)
06200           WRITE(5,50)JCTLFL
06300   50     FORMAT(' CONTROL DATA FILE IS --- ',A10)
06400   C-----------------------------------------------------------------------
06500   C
```

```
C-----------------------------------------------------------------
C
      READ(23,250)NDATA,NPAR,ISERCH,IOP,ISENSE,IMODEL,IREPAR
      READ(23,260)UM,UK,BM,BK,Z,X
      READ(23,261)GAMA,PHI1
      READ(23,270)USI,BSI
      READ(23,272)ITMAX,INPOL,PAREPS,FNEPS,STPMAX
      READ(23,274)PLM,PLMAX,PLMULT,PLDIVD
C
      WRITE(5,449)
      WRITE(5,450)NDATA,NPAR,ISERCH,IOP,ISENSE,IMODEL,IREPAR
      WRITE(5,459)
      WRITE(5,460)UM,UK,BM,BK,Z,X
      WRITE(5,461)GAMA,PHI1
      WRITE(5,469)
      WRITE(5,470)USI,BSI
      WRITE(5,471)
      WRITE(5,472)ITMAX,INPOL,PAREPS,FNEPS,STPMAX
      WRITE(5,474)PLM,PLMAX,PLMULT,PLDIVD
      WRITE(5,479)
C
250   FORMAT(5X,7(I4,X))
260   FORMAT(5X,6(F10.0,X))
261   FORMAT(5X,2(F10.0,X))
270   FORMAT(5X,2(F10.0,X))
272   FORMAT(5X,2(I4,X),3(F12.0,X))
274   FORMAT(5X,4(F12.0,X))
C
449   FORMAT(3X,'-------- CONTROL INFORMATION --------')
450   FORMAT(3X,'NDATA=',I4,' NPAR=',I4,' ISERCH=',I4,' IOP=',
     1  I4,' ISENSE=',I4,' IMODEL=',I4,' IREPAR=',I4)
459   FORMAT(3X,'-------- INITIAL PAR VALUES --------')
460   FORMAT(3X,'UM=',F10.5,5X,'UK=',F10.5,/,3X,'BM=',F10.5,
     1  5X,'BK=',F10.5,/,3X,' Z=',F10.5,5X,' X=',F10.5)
461   FORMAT(3X,'GAMA=',F10.5,3X,'PHI1=',F10.5)
469   FORMAT(3X,'-------- INITIAL STATE VALUES --------')
470   FORMAT(3X,'USI=',F10.5,5X,'BSI=',F10.5)
471   FORMAT(3X,'-------- GRADIENT SEARCH INFO --------')
472   FORMAT(3X,'ITMAX=',I4,' INPOL=',I4,' PAREPS=',G12.6,
     1  ' FNEPS=',G12.6,' STPMAX=',G12.6)
474   FORMAT(3X,'PLM=',G12.6,' PLMAX=',G12.6,' PLMULT=',G12.6,
     1  'PLDIVD=',G12.6)
479   FORMAT(3X,50('-'))
C
C
C
C
C-----------------------------------------------------------------
C
C   READ PPT. INPUT DATA FILE
C
      DO 310 I=1,NDATA
      READ(20,300)P(I)
300   FORMAT(F10.0)
310   CONTINUE
C
C-----------------------------------------------------------------
C
C     READ OBSERVED FLOWS FILE
C
      DO 305 I=1,NDATA
      READ(21,306)OBSQ(I)
306   FORMAT(F20.0)
305   CONTINUE
C
C-----------------------------------------------------------------
C
C-----------------------------------------------------------------
C
      RETURN
      END
```

```
C ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
C
        SUBROUTINE NOSRCH
C
C------------------------------------------------------------
C
C  ***  CONTROLS THE OPERATIONS OF THE NON-OPTIMIZATION RUNS
C
C------------------------------------------------------------
C
        COMMON/CNTROL/ISERCH,IOP,ISENSE,IMODEL,IREPAR
        COMMON/ESTIM/PARVAL(10),SLOP(10),HESS(10,10)
        COMMON/PARAMS/NPAR,UM,UK,SM,SK,Z,X,USI,BSI
        COMMON/REPAR/A,RKU,RKB,XINV,AINV
C
C-------------------------------------------------------------
C
        WRITE(5,100)
100     FORMAT(5X,'*** SUBROUTINE NOSRCH ***',/)
C
C-------------------------------------------------------------
C
        IF(IMODEL.EQ.2)GO TO 200
C
        WRITE(5,105)
105     FORMAT(5X,'*** START MODELZ ***',/)
        CALL MODELZ
        GO TO 300
C
200     CONTINUE
        WRITE(5,110)
110     FORMAT(5X,'*** START MODELA ***',/)
        CALL MODELA
C
300     CONTINUE
C
        IF(IREPAR.NE.1)GO TO 600
        WRITE(5,115)
115     FORMAT(5X,'*** START PARIZ3 ***',/)
        CALL PARIZ3
C
600     CONTINUE
C
C--------------------------------------------------------------
C
        IF(IOP.EQ.2) GO TO 666
C
C-------------------------------------------------------------
C
C  ***  HESSIAN FOR SLS FUNCTION
C
        WRITE(5,120)
120     FORMAT(5X,'*** START DERIV2 ***',/)
        CALL DERIV2
C
        GO TO 700
C
C-------------------------------------------------------------
C
666     CONTINUE
C
C  ***  HESSIAN FOR NLL FUNCTION BASED ON THE RATING CURVE MODEL
C
        WRITE(5,610)
610     FORMAT(5X,'*** START LOGLIK ***')
        CALL LOGLIK(FUNC)
        WRITE(5,620)
620     FORMAT(5X,'*** START DERIV1 ***')
        CALL DERIV1
        WRITE(5,630)
630     FORMAT(5X,'*** START DERIV2 ***')
        CALL DERIV2
C
C-------------------------------------------------------------
C
700     CONTINUE
C
C
```

```
C-----------------------------------------------------------------
C
      IF(ISENSE.NE.1)GO TO 400
C
      WRITE(5,125)
125   FORMAT(5X,'*** START PARIZ1 ***',/)
      CALL PARIZ1
C
C
      WRITE(5,130)
130   FORMAT(5X,'*** START ANLYS2 ***',/)
      CALL ANLYS2
C
400   CONTINUE
C
C-----------------------------------------------------------------
C
C**********************
9999  CONTINUE
C**********************
C
      RETURN
      END
```

```
CZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
C
      SUBROUTINE GSERCH
C
C-----------------------------------------------------------------
C
C  ***  CONTROLS THE OPERATION OF THE LEVENBERG-MARQUARDT
C  ***  MODIFIED NEWTON TYPE GRADIENT SEARCH ALGORITHM
C  ***  OPTIMIZATION RUNS.
C
C-----------------------------------------------------------------
C
      COMMON/CNTROL/ISERCH, IOP, ISENSE, IMODEL, IREPAR
      COMMON/PARAMS/NPAR, UM, UK, BM, BK, Z, X, USI, BSI
      COMMON/REPAR/A, RKU, RKB, XINV, AINV
      COMMON/ESTIM/PARVAL(10), SLOP(10), HESS(10,10)
      COMMON/SERCH1/ITMAX, INPOL, PAREPS, FNEPS, STPMAX, PLM, PLMAX,
     1 PLMULT, PLDIVD
C
C
C------------------------------------------------------------------
C
      WRITE(5,100)
100   FORMAT(3X,'*** SUBROUTINE GSERCH ***',/,50('-'),/,3X,
     1 '*** LEVENBERG-MARQUARDT GRADIENT SEARCH ALGORITHM ***',
     2 /,50('-'),/)
C
C------------------------------------------------------------------
C
      WRITE(5,110)
110   FORMAT(3X,'*** START PARIZ1 ***',/)
C
      CALL PARIZ1
C
C------------------------------------------------------------------
C
      WRITE(5,120)
120   FORMAT(3X,'*** START GMAIN ***',/)
C
      CALL GMAIN(NPAR,PARVAL,ITMAX,PAREPS,FNEPS,INPOL,
     1 PLM,PLMAX,PLMULT,PLDIVD,STPMAX)
C
C------------------------------------------------------------------
C
      IF(ISENSE.NE.1)GO TO 200
C
      WRITE(5,130)
130   FORMAT(3X,'*** START ANLYS2 ***',/)
C
      CALL ANLYS2
C
C------------------------------------------------------------------
C
200   CONTINUE
C
      RETURN
      END
```

```
00100   C2222222222222222222222222222222222222222222222222222222222222222222222
00200   C
00300           SUBROUTINE DSERCH
00400   C
00500   C-----------------------------------------------------------------------
00600   C
00700   C  ***  CONTROLS THE OPERATION OF THE DIRECT SEARCH OPTIMIZATION
00800   C  ***  RUNS USING THE SIMPLEX NONLINEAR ALGORITHM OF NELDER AND
00900   C  ***  MEAD (1966).
01000   C
01100   C-----------------------------------------------------------------------
01200   C
01300           COMMON/CNTROL/ISERCH, IOP, ISENSE, IMODEL, IREPAR
01400           COMMON/PARAMS/NPAR, UM, UK, BM, BK, Z, X, USI, BSI
01600           COMMON/ESTIM/PARVAL(10), SLOP(10), HESS(10,10)
01700           COMMON/SERCH1/ITMAX, INPOL, PAREPS, FNEPS, STPMAX, PLM, PLMAX,
01800          1  PLMULT, PLDIVD
01900   C
02000   C
02100   C-----------------------------------------------------------------------
02200   C
02300           WRITE(5,100)
02400   100     FORMAT(3X, '*** SUBROUTINE DSERCH ***',/,50('-'),/,3X,
02500          1  '*** SIMPLEX DIRECT SEARCH ALGORITHM ***',/,50('-'),/)
02600   C
02700   C-----------------------------------------------------------------------
02800   C
02900           WRITE(5,110)
03000   110     FORMAT(3X, '*** START PARIZ1 ***',/)
03100   C
03200           CALL PARIZ1
03300   C
03400   C-----------------------------------------------------------------------
03500   C
03600           WRITE(5,120)
03700   120     FORMAT(3X, '*** START SIMPLX ***',/)
03800   C
03900           CALL SIMPLX(NPAR, ITMAX, PAREPS, FNEPS, PARVAL)
04000   C
04100   C-----------------------------------------------------------------------
05000   C-----------------------------------------------------------------------
05100   C
05300   C
05400           RETURN
05500           END
```

```
00100   C-----------------------------------------------------------
00200   C-----------------------------------------------------------
00300   C
00400   C      6-PARAMETER MODEL BASED ON BURNASH MODEL STRUCTURE
00500   C
00600             SUBROUTINE MODEL2
00700   C
00800   C---------------------- ----------------------------------------
00900   C
01000             COMMON/RINPUT/NDATA,P(999),OBSQ(999),TRUQ(999)
01100             COMMON/ROUTPT/Q(999),R(999),S(999),B(999),PPERC(999),PERC(999)
01200             COMMON/STATES/US(999),BS(999)
01250             COMMON/PARAMS/NPAR,UM,UK,BM,BK,Z,X,USI,BSI
01300   C
01400             COMMON/ESTIM2/DQ(999,6)
01500   C
01600   C-----------------------------------------------------------------
01700   C
01800   C
01900   C
02000             I=1
02100             US(I)=USI
02200             BS(I)=BSI
02300   C
02400             DUSUM=0.0
02500             DUSUK=0.0
02600             DUSBM=0.0
02700             DUSBK=0.0
02800             DUSZ =0.0
02900             DUSX =0.0
03000   C
03100             DBSUM=0.0
03200             DBSUK=0.0
03300             DBSBM=0.0
03400             DBSBK=0.0
03500             DBSZ=0.0
03600             DBSX=0.0
03700   C
03800   C
03900   90        CONTINUE
04000             R(I)=0.0
04100             S(I)=0.0
04200             B(I)=0.0
04300             US(I)=US(I)+P(I)
04400   C-----------------------------------------------------
04500             YY=BM+BK+US(I)/UM
04600             CC=Z*( (BM-BS(I))/BM )**X
04700   C
04800             PPERC(I)=YY+CC*YY
04900   C
05000             IF(PPERC(I).LT.US(I))GO TO 900
05100   C
05200             PERC(I)=US(I)
05300   C
05400             DPUM=DUSUM
05500             DPUK=DUSUK
05600             DPBM=DUSBM
05700             DPBK=DUSBK
05800             DPZ =DUSZ
05900             DPX =DUSX
06000   C
06100             US(I)=0.0
06200   C
06300             DUSUM=0.0
06400             DUSUK=0.0
06500             DUSBM=0.0
06600             DUSBK=0.0
06700             DUSZ=0.0
06800             DUSX=0.0
06900   C
07000             GO TO 910
07100   C
07200   900       CONTINUE
07300   C
```

```
07400   C
07500           PERC(I)=PERC(I)
07600   C
07700           DYYUM=BM**EX*DUSUM/UM - YY/UM
07800           DYYUK=BM**EX*DUSUK/UM
07900           DYYBM=BM**EX*DUSBM/UM - YY/BM
08000           DYYBK=BM**EX*DUSBK/UM - YY/BK
08100           DYYZ =BM**EX*DUSZ/UM
08200           DYYX =BM**EX*DUSX/UM
08300   C
08400           CAC= (X*CC**( (X-1)/X ) )*( Z**(1.0/X) )
08500   C
08600           DCCUM= -DBSUM*CAC/BM
08700           DCCUK= -DBSUK*CAC/BM
08800           DCCBM= -CAC*( DBSBM/BM - BS(I)/(BM**2) )
08900           DCCBK= -CAC*DBSBK/BM
09000           DCCZ = CC/Z - CAC*DBSZ/BM
09100           IF(BM.NE.BS(I))GO TO 65.
09200           DCCX=0.0
09300           GO TO 67
09400   65      CONTINUE
09500           DCCX=CC*(ALOG((BM-BS(I))/BM) - (X*DBSX)/(BM-BS(I)) )
09600   67      CONTINUE
09700   C
09800           DPUM = DYYUM*(1.0-CC) -YY*DCCUM
09900           DPUK = DYYUK*(1.0-CC) - YY*DCCUK
10000           DPBM = DYYBM*(1.0-CC) - YY*DCCBM
10100           DPBK = DYYBK*(1.0-CC) - YY*DCCBK
10200           DPZ  = DYYZ*(1.0-CC) -YY*DCCZ
10300           DPX  = DYYX*(1.0-CC) - YY*DCCX
10400   C
10500           US(I)=US(I)-PERC(I)
10600   C
10700           DUSUM=DUSUM-DPUM
10800           DUSUK=DUSUK-DPUK
10900           DUSBM=DUSBM-DPBM
11000           DUSBK=DUSBK-DPBK
11100           DUSZ=DUSZ-DPZ
11200           DUSX=DUSX-DPX
11300   C
11400   910     CONTINUE
11500   C
11600   C*****************(*****************************
11700            BS(I)=BS(I)+PERC(I)
11800   C
11900           DBSUM=DBSUM+DPUM
12000           DBSUK=DBSUK+DPUK
12100           DBSBM=DBSBM+DPBM
12200           DBSBK=DBSBK+DPBK
12300           DBSZ =DBSZ+DPZ
12400           DBSX =DBSX+DPX
12500   C
12600           IF(BS(I).GT.BM)GO TO 120
12700             .
12800           B(I)=BS(I)*BK
12900   C
13000           DBUM = DBSUM*BK
13100           DBUK = DBSUK*BK
13200           DBBM = DBSBM*BK
13300           DBBK = DBSBK*BK +BS(I)
13400           DBZ  = DBSZ*BK
13500           DBX  = DBSX*BK
13600   C
13700           BS(I)=BS(I)-B(I)
13800   C
13900           DBSUM=DBSUM-DBUM
14000           DBSUK=DBSUK-DBUK
14100           DBSBM=DBSBM-DBBM
14200           DBSBK=DBSBK-DBBK
14300           DBSZ =DBSZ -DBZ
14400           DBSX =DBSX -DBX
14500   C
14600           GO TO 130
14700   120     CONTINUE
14800
```

```
14900              D=BS(I)-BM
15000    C
15100              DDUM=DBSUM
15200              DDUK=DBSUK
15300              DDBM=DBSBM-1.0
15400              DDBK=DBSBK
15500              DDZ =DBSZ
15600              DDX = DBSX
15700    C
15800              B(I)=BM+BK
15900    C
16000              DBUM=0.0
16100              DBUK=0.0
16200              DBBM=BK
16300              DBBK=BM
16400              DBZ =0.0
16500              DBX=0.0
16600    C
16700              BS(I)=BM-B(I)+D
16800    C
16900    C
17000              DBSUM=-DBUM+DDUM
17100              DBSUK=-DBUK+DDUK
17200              DBSBM=1.0-DBBM+DDBM
17300              DBSBK=-DBBK+DDBK
17400              DBSZ =-DBZ +DDZ
17500              DBSX =-DBX +DDX
17600    C
17700              IF(BS(I).GT.BM)GO TO 140
17800              GO TO 150
17900    140       CONTINUE
18000              D=BS(I)-BM
18100    C
18200              DDUM=DBSUM
18300              DDUK=DBSUK
18400              DDBM=DBSBM-1.0
18500              DDBK=DBSBK
18600              DDZ =DBSZ
18700              DDX = DBSX
18800    C
18900              BS(I)=BM
19000    C
19100              DBSUM=0.0
19200              DBSUK=0.0
19300              DBSBM=1.0
19400              DBSBK=0.0
19500              DBSZ =0.0
19600              DBSX =0.0
19700    C
19800              US(I)=US(I)+D
19900    C
20000              DUSUM=DUSUM+DDUM
20100              DUSUK=DUSUK+DDUK
20200              DUSBM=DUSBM+DDBM
20300              DUSBK=DUSBK+DDBK
20400              DUSZ =DUSZ+DDZ
20500              DUSX =DUSX+DDX
20600    C
20700    150       CONTINUE
20800    130       CONTINUE
20900    C
21000              S(I)=US(I)+UK
21100    C
21200              DSUM=DUSUM+UK
21300              DSUK=DUSUK+UK+US(I)
21400              DSBM=DUSBM+UK
21500              DSBK=DUSBK+UK
21600              DSZ =DUSZ+UK
21700              DSX =DUSX+UK
21800    C
21900              US(I)=US(I)-S(I)
22000    C
22100              DUSUM=DUSUM-DSUM
22200              DUSUK=DUSUK-DSUK
22300              DUSBM=DUSBM-DSBM
22400              DUSBK=DUSBK-DSBK
22500              DUSZ =DUSZ -DSZ
22600              DUSX =DUSX -DSX
22700    C
```

```
22800           IF(US(I) GT.UM)GO TO 160
22900   C
23000           R(I)=0.0
23100   C
23200           DRUM=0.0
23300           DRUK=0.0
23400           DRBM=0.0
23500           DRBK=0.0
23600           DRZ=0.0
23700           DRX=0.0
23800   C
23900           GO TO 170
24000   160     CONTINUE
24100   C
24200           R(I)=US(I)-UM
24300   C
24400           DRUM=DUSUM-1.0
24500           DRUK=DUSUK
24600           DRBM=DUSBM
24700           DRBK=DUSBK
24800           DRZ =DUSZ
24900           DRX =DUSX
25000   C
25100           US(I)=UM
25200   C
25300           DUSUM=1.0
25400           DUSUK=0.0
25500           DUSBM=0.0
25600           DUSBK=0.0
25700           DUSZ=0.0
25800           DUSX=0.0
25900   C
26000   170     CONTINUE
26100   C
26200           G(I)=R(I)+S(I)+B(I)
26300   C
26400           DG(I,1)=DRUM+DSUM+DBUM
26500           DG(I,2)=DRUK+DSUK+DBUK
26600           DG(I,3)=DRBM+DSBM+DBBM
26700           DG(I,4)=DRBK+DSBK+DBBK
26800           DG(I,5) =DRZ +DSZ +DBZ
26900           DG(I,6) =DRX +DSX +DBX
26910   C
26940   C
27000   C
27100   C
27200   C
27300   C
27400           IF(I.GE.NDATA)GO TO 180
27500           US(I+1)=US(I)
27600           BS(I+1)=BS(I)
27700           I=I+1
27800           GO TO 90
27900   C
28000   C---------------------------------------------------------------
28100   C
28200   180     CONTINUE
28300   C
28400   C
28500   C
28600           RETURN
28700           END
```

```
00100   CZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
00200   C-------------------------------------------------------------------
00300   C
00400   C      6-PARAMETER MODEL BASED ON BURNASH MODEL STRUCTURE
00500   C
00600             SUBROUTINE MODL21
00700   C
00710   C   ***  COMPUTES ONLY OUTPUT VALUES AND NOT DERIVATIVES
00720   C
00800   C-------------------------------------------------------------------
00900   C
01000             COMMON/RINPUT/NDATA,P(999),OBSQ(999),TRUQ(999)
01100             COMMON/ROUTPT/Q(999),R(999),S(999),B(999),PPERC(999),PERC(999)
01200             COMMON/STATES/US(999),BS(999)
01250             COMMON/PARAMS/NPAR,UM,UK,BM,BK,I,X,USI,BSI
01300   C
01600   C-------------------------------------------------------------------
01700   C
02000             I=1
02100             US(I)=USI
02200             BS(I)=BSI
02300   C
03900   90        CONTINUE
04000             R(I)=0.0
04100             S(I)=0.0
04200             B(I)=0.0
04300             US(I)=US(I)+P(I)
04400   C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
04500             YY=BM*BK*US(I)/UM
04600             CC=I*( (BM-BS(I))/BM )**X
04700   C
04800             PPERC(I)=YY+CC*YY
04900   C
05000             IF(PPERC(I).LT.US(I))GO TO 900
05100   C
05200             PERC(I)=US(I)
06100             US(I)=0.0
07000             GO TO 910
07100   C
07200   900       CONTINUE
07300   C
07500             PERC(I)=PPERC(I)
10500             US(I)=US(I)-PERC(I)
10600   C
11400   910       CONTINUE
11500   C
11600   C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
11700              BS(I)=BS(I)+PERC(I)
11800   C
12600             IF(BS(I).GT.BM)GO TO 120
12700   C
12800             B(I)=BS(I)*BK
13700             BS(I)=BS(I)-B(I)
14600             GO TO 130
14650   C
14700   120       CONTINUE
14800   C
14900             D=BS(I)-BM
15800             B(I)=BM*BK
16700             BS(I)=BM-B(I)+D
16800   C
17700             IF(BS(I).GT.BM)GO TO 140
17800             GO TO 150
17900   140       CONTINUE
18000             D=BS(I)-BM
18900             BS(I)=BM
19800             US(I)=US(I)+D
19900   C
20700   150       CONTINUE
20800   130       CONTINUE
20900   C
```

```
21000           S(I)=US(I)+UM
21900           US(I)=US(I)-S(I)
22000   C
22800           IF(US(I).GT.UM)GO TO 160
22900   C
23000           R(I)=0.0
23100   C
23900           GO TO 170
24000   160     CONTINUE
24100   C
24200           R(I)=US(I)-UM
25100           US(I)=UM
25200   C
26000   170     CONTINUE
26100   C
26200           Q(I)=R(I)+S(I)+B(I)
26300   C
27400           IF(I.GE.NDATA)GO TO 180
27500           US(I+1)=US(I)
27600           BS(I+1)=BS(I)
27700           I=I+1
27800           GO TO 90
27900   C
28000   C-------------------- ----- ------------------------------
28100   C
28200   180     CONTINUE
28300   C
28400   C
28500   C
28600           RETURN
28700           END
```

```
00100   C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
00200   C---------------------------------------------------------------
00300   C
00400   C      6-PARAMETER MODEL BASED ON BURNASH MODEL STRUCTURE
00500   C
00600           SUBROUTINE MODEL6
00700   C
00800   C---------------------------------------------------------------
00900   C
01000           COMMON/RINPUT/NDATA,P(999),OBSQ(999),TRUQ(999)
01100           COMMON/ROUTPT/Q(999),R(999),S(999),B(999),PPERC(999),PERC(999)
01200           COMMON/STATES/US(999),BS(999)
01300           COMMON/PARAMS/NPAR,UM,UK,BM,BK,Z,X,USI,BSI
01400           COMMON/REPAR/A,RKU,RKB,XINV,A1MV
01500   C
01600           COMMON/ESTIM2/DQ(999,6)
01700   C
01800   C---------------------------------------------------------------
01900   C
02000   C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
02100   C
02200           A = Z
02700   C
02800   C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
02900   C
03000   C
03100           I=1
03200           US(I)=USI
03300           BS(I)=BSI
03400   C
03500           DUSUM=0.0
03600           DUSUK=0.0
03700           DUSBM=0.0
03800           DUSBK=0.0
03900           DUSA =0.0
04000           DUSX =0.0
04100   C
04200           DESUM=0.0
04300           DESUK=0.0
04400           DESBM=0.0
04500           DESBK=0.0
04600           DESA=0.0
04700           DESX=0.0
04800   C
04900   C
05000   90      CONTINUE
05100           R(I)=0.0
05200           S(I)=0.0
05300           B(I)=0.0
05400           US(I)=US(I)+P(I)
05500   C+++++++++++++++++++++++++++++++++++++++++++++++++
05600   C
05700           RLZDR=( BM-BS(I) )/BM
05800   C
05900           IF( RLZDR.LE.A )GO TO 900
06000   C
06100           PERC(I)=US(I)
06200   C
06300           DPUM=DUSUM
06400           DPUK=DUSUK
06500           DPBM=DUSBM
06600           DPBK=DUSBK
06700           DPA =DUSA
06800           DPX =DUSX
06900   C
07000           US(I)=0.0
07100   C
07200           DUSUM=0.0
07300           DUSUK=0.0
07400           DUSBM=0.0
07500           DUSBK=0.0
07600           DUSA=0.0
07700           DUSX=0.0
07800   C
07900           GO TO 910
08000   C
08100   900     CONTINUE
08200   C
```

```
08300            YY=US(I)*BM*BK/UM
08400            ZZ=( ( BM-BS(I))/(BM*A) )**X
08500    C
08600            PERC(I)=YY + ZZ*( US(I)-YY )
08700    C
08800            DYYUM=BM*BK*DUSUM/UM - YY/UM
08900            DYYUK=BM*BK*DUSUK/UM
09000            DYYBM=BM*BK*DUSBM/UM + YY/BM
09100            DYYBK=BM*BK*DUSBK/UM + YY/BK
09200            DYYA =BM*BK*DUSA/UM
09300            DYYX =BM*BK*DUSX/UM
09400    C
09420            IF(BS(I).EQ.BM)GO TO 62
09500            ZAZ= X*ZZ**( (X-1)/X )
09510            GO TO 63
09520    62      CONTINUE
09530            ZAZ=0.0
09540    63      CONTINUE
09600    C
09700            DZZUM= -DBSUM*ZAZ/(BM*A)
09800            DZZUK= -DBSUK*ZAZ/(BM*A)
09900            DZZBM= -(ZAZ/A)*( DBSBM/BM - BS(I)/(BM**2) )
10000            DZZBK= -(ZAZ/A)*DBSBK/BM
10100            DZZA = -(ZAZ/A)*( (BM-BS(I))/(BM*A) + DBSA/BM )
10200            IF(BM.NE.BS(I))GO TO 65
10300            DZZX=0.0
10400            GO TO 67
10500    65      CONTINUE
10600            DZZX=ZZ*(ALOG((BM-BS(I))/(BM*A)) - (X*DBSX)/(BM-BS(I)) )
10700    67      CONTINUE
10800    C
10900            DPUM = DYYUM + DZZUM*(US(I)-YY) + ZZ*(DUSUM-DYYUM)
11000            DPUK = DYYUK + DZZUK*(US(I)-YY) + ZZ*(DUSUK-DYYUK)
11100            DPBM = DYYBM + DZZBM*(US(I)-YY) + ZZ*(DUSBM-DYYBM)
11200            DPBK = DYYBK + DZZBK*(US(I)-YY) + ZZ*(DUSBK-DYYBK)
11300            DPA  = DYYA  + DZZA*(US(I)-YY)  + ZZ*(DUSA-DYYA)
11400            DPX  = DYYX  + DZZX*(US(I)-YY)  + ZZ*(DUSX-DYYX)
11500    C
11600            US(I)=US(I)-PERC(I)
11700    C
11800            DUSUM=DUSUM-DPUM
11900            DUSUK=DUSUK-DPUK
12000            DUSBM=DUSBM-DPBM
12100            DUSBK=DUSBK-DPBK
12200            DUSA=DUSA-DPA
12300            DUSX=DUSX-DPX
12400    C
12500    910     CONTINUE
12600    C
12700    C*********-----------------------------------------
12800             BS(I)=BS(I)+PERC(I)
12900    C
13000            DBSUM=DBSUM+DPUM
13100            DBSUK=DBSUK+DPUK
13200            DBSBM=DBSBM+DPBM
13300            DBSBK=DBSBK+DPBK
13400            DBSA =DBSA+DPA
13500            DBSX =DBSX+DPX
13600    C
13700            IF(BS(I).GT.BM)GO TO 120
13800
13900            B(I)=BS(I)*BK
14000    C
14100            DBUM = DBSUM*BK
14200            DBUK = DBSUK*BK
14300            DBBM = DBSBM*BK
14400            DBBK = DBSBK*BK +BS(I)
14500            DBA  = DBSA*BK
14600            DBX  = DBSX*BK
14700    C
14800            BS(I)=BS(I)-B(I)
14900    C
15000            DBSUM=DBSUM-DBUM
15100            DBSUK=DBSUK-DBUK
15200            DBSBM=DBSBM-DBBM
15300            DBSBK=DBSBK-DBBK
15400            DBSA =DBSA -DBA
15500            DBSX =DBSX -DBX
15600    C
```

```
15700                GO TO 130
15800     120        CONTINUE
15900
16000                D=BS(I)-BM
16100     C
16200                DDUM=DBSUM
16300                DDUK=DBSUK
16400                DDBM=DBSBM-1.0
16500                DDBK=DBSBK
16600                DDA =DBSA
16700                DDX = DBSX
16800     C
16900                B(I)=BM+BK
17000     C
17100                DBUM=0.0
17200                DBUK=0.0
17300                DBBM=BK
17400                DBBK=BM
17500                DBA =0.0
17600                DBX=0.0
17700     C
17800                BS(I)=BM-B(I)+D
17900     C
18000     C
18100                DBSUM=-DBUM+DDUM
18200                DBSUK=-DBUK+DDUK
18300                DBSBM=1.0-DBBM+DDBM
18400                DBSBK=-DBBK+DDBK
18500                DBSA =-DBA +DDA
18600                DBSX =-DBX +DDX
18700     C
18800                IF(BS(I).GT.BM)GO TO 140
18900                GO TO 150
19000     140        CONTINUE
19100                D=BS(I)-BM
19200     C
19300                DDUM=DBSUM
19400                DDUK=DBSUK
19500                DDBM=DBSBM-1.0
19600                DDBK=DBSBK
19700                DDA =DBSA
19800                DDX = DBSX
19900     C
20000                BS(I)=BM
20100     C
20200                DBSUM=0.0
20300                DBSUK=0.0
20400                DBSBM=1.0
20500                DBSBK=0.0
20600                DBSA =0.0
20700                DBSX =0.0
20800     C
20900                US(I)=US(I)+D
21000     C
21100                DUSUM=DUSUM+DDUM
21200                DUSUK=DUSUK+DDUK
21300                DUSBM=DUSBM+DDBM
21400                DUSBK=DUSBK+DDBK
21500                DUSA =DUSA+DDA
21600                DUSX =DUSX+DDX
21700     C
21800     150        CONTINUE
21900     130        CONTINUE
22000     C
22100                S(I)=US(I)*UK
22200     C
22300                DSUM=DUSUM*UK
22400                DSUK=DUSUK*UK+US(I)
22500                DSBM=DUSBM*UK
22600                DSBK=DUSBK*UK
22700                DSA =DUSA*UK
22800                DSX =DUSX*UK
22900     C
23000                US(I)=US(I)-S(I)
23100     C
23200                DUSUM=DUSUM-DSUM
23300                DUSUK=DUSUK-DSUK
23400                DUSBM=DUSBM-DSBM
23500                DUSBK=DUSBK-DSBK
23600                DUSA =DUSA -DSA
23700                DUSX =DUSX -DSX
```

```
23800   C
23900           IF(US(I).GT.UM)GO TO 160
24000   C
24100           R(I)=0.0
24200   C
24300           DRUM=0.0
24400           DRUK=0.0
24500           DRBM=0.0
24600           DRBK=0.0
24700           DRA=0.0
24800           DRI=0.0
24900   C
25000           GO TO 170
25100   160     CONTINUE
25200   C
25300           R(I)=US(I)-UM
25400   C
25500           DRUM=DUSUM-1.0
25600           DRUK=DUSUK
25700           DRBM=DUSBM
25800           DRBK=DUSBK
25900           DRA =DUSA
26000           DRI =DUSI
26100   C
26200           US(I)=UM
26300   C
26400           DUSUM=1.0
26500           DUSUK=0.0
26600           DUSBM=0.0
26700           DUSBK=0.0
26800           DUSA=0.0
26900           DUSI=0.0
27000   C
27100   170     CONTINUE
27200   C
27300           Q(I)=R(I)+S(I)+B(I)
27310   C----------------
27320   C       WRITE(5,567)Q(I)
27330   C  567  FORMAT(F20.14)
27335   C
27340   C----------------------
27400   C
27500           DQ(I,1)=DRUM+DSUM+DBUM
27600           DQ(I,2)=(DRUK+DSUK+DBUK)
27700           DQ(I,3)=DRBM+DSBM+DBBM
27800           DQ(I,4)=( DRBK+DSBK+DBBK)
27900           DQ(I,5) =(DRA +DSA +DBA)
28000           DQ(I,6) =(DRI +DSI +DBI)
28100   C
28200   C
28300   C
28400           IF(I.GE.NDATA)GO TO 180
28500           US(I+1)=US(I)
28600           BS(I+1)=BS(I)
28700           I=I+1
28800           GO TO 90
28900   C
29000   C-------------------------------------------------------------
29100   C
29200   180     CONTINUE
29300   C
29400           RETURN
29500           END
```

```
29600   C-------------------- ----------------------------------------
29700   C-------------------------------------------------------------
29800   C
29900   C      6-PARAMETER MODEL BASED ON SURNASH MODEL STRUCTURE
30000   C
30100           SUBROUTINE MODLA1
30200   C
30300   C-------------------------------------------------------------
30400   C
30500           COMMON/RINPUT/NDATA,P(999),OBSQ(999),TRUQ(999)
30600           COMMON/ROUTPT/Q(999),R(999),S(999),B(999),PPERC(999),PERC(999)
30700           COMMON/STATES/US(999),BS(999)
30800           COMMON/PARAMS/NPAR,UM,UK,BM,BK,Z,X,USI,BSI
30900           COMMON/REPAR/A,RM,U,RKB,XINV,AINV
31000   C
31100   C-------------------------------------------------------------
31200   C
31300   C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
31400   C
31500           A = Z
31600   C
31700   C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
31800   C
31900   C
32000           I=1
32100           US(I)=USI
32200           BS(I)=BSI
32300   C
32400   90      CONTINUE
32500           R(I)=0.0
32600           S(I)=0.0
32700           B(I)=0.0
32800           US(I)=US(I)+P(I)
32900   C+++++++++++++++++++++++++++++++++++++++++++++++
33000   C
33100           RLIDR=( BM-BS(I) )/BM
33200   C
33300           IF( RLIDR.LE.A )GO TO 900
33400   C
33500           PERC(I)=US(I)
33600           US(I)=0.0
33700           GO TO 910
33800   C
33900   900     CONTINUE
34000   C
34100           YY=US(I)*BM*BK/UM
34200           ZZ=( ( BM-BS(I))/(BM*A) )**X
34300   C
34400           PERC(I)=YY + ZZ*( US(I)-YY )
34500           US(I)=US(I)-PERC(I)
34600   910     CONTINUE
34700   C
34800   C+++++++++++++++++++++++++++++++++++++++++++++++++
34900            BS(I)=BS(I)+PERC(I)
35000   C
35100           IF(BS(I).GT.BM)GO TO 120
35200
35300           B(I)=BS(I)*BK
35400           BS(I)=BS(I)-B(I)
35500           GO TO 130
35600   120     CONTINUE
35700
35800           D=BS(I)-BM
35900           B(I)=BM*BK
36000           BS(I)=BM-B(I)+D
36100   C
36200           IF(BS(I).GT.BM)GO TO 140
36300           GO TO 150
36400   140     CONTINUE
36500           D=BS(I)-BM
36600           BS(I)=BM
36700           US(I)=US(I)+D
36800   150     CONTINUE
36900   130     CONTINUE
37000   C
37100           S(I)=US(I)*UK
37200           US(I)=US(I)-S(I)
```

```
C
      IF(US(I).GT.UM)GO TO 160
C
      R(I)=0.0
C
      GO TO 170
160   CONTINUE
C
      R(I)=US(I)-UM
      US(I)=UM
170   CONTINUE
C
      Q(I)=R(I)+S(I)+B(I)
C
C
      IF(I.GE.NDATA)GO TO 180
      US(I+1)=US(I)
      BS(I+1)=BS(I)
      I=I+1
      GO TO 90
C
C-----------------------------------------------------------------
C
180   CONTINUE
C
      RETURN
      END
```

```
00100   C
00200           SUBROUTINE LOGLIK(FUNC)
00300   C
00400   C-----------------------------------------------------------------
00500   C
00600           COMMON/RINPUT/NDATA,P(999),OBSQ(999),TRUQ(999)
00700           COMMON/ROUTPT/Q(999),R(999),S(999),B(999),PPERC(999),PERC(999)
00800           COMMON/CNTROL/ISERCH,IOP,ISENSE,IMODEL,IREPAR
00850           COMMON/PARAM2/GAMA,PHI1,SIGSQR
00875           COMMON/ERROR/V(999),EPS(999)
00900   C
01000   C-----------------------------------------------------------------
01020   C
01040           IF(IOP.EQ.2)GO TO 200
01060   C
01080   C-----------------------------------------------------------------
01100   C
01200   C  ***  SIMPLE LEAST SQUARES FUNCTION
01300   C
01350           SUM=0.0
01400           DO 100 I=1,NDATA
01500           SUM=SUM + ( Q(I)-OBSQ(I) )*( Q(I)-OBSQ(I) )
01600   100     CONTINUE
01700   C
01800           FUNC=SUM
01820   C
01840           GO TO 999
01900   C
02000   C-----------------------------------------------------------------
02020   C
02040   200     CONTINUE
02100   C
02200   C  ***  NEGATIVE LOGLIKELIHOOD BASED ON THE RATING CURVE MODEL
02300   C
02400           SUM1=0.0
02500           SUM2=0.0
02600           V(1)=( OBSQ(1)**GAMA - Q(1)**GAMA ) / GAMA
02700   C
02800           DO 300 I=2,NDATA
02900           V(I)=( OBSQ(I)**GAMA - Q(I)**GAMA ) / GAMA
03000           EPS(I)=V(I)-PHI1*V(I-1)
03100           SUM1=SUM1+EPS(I)*EPS(I)
03200           SUM2=SUM2+ALOG( OBSQ(I) )
03300   300     CONTINUE
03400   C
03450           DATA=NDATA
03500           SIGSQR=SUM1/(DATA-1.0)
03600   C
03700           RA = (DATA-1.0)*1.8382795/2.0
03800           RB = (DATA-1.0)*ALOG(SIGSQR)/2.0
03900           RC = (1.0-GAMA)*SUM2
04000           RD = (DATA-1.0)/2.0
04100   C
04200           FUNC = RA + RB + RC + RD
04300   C
04400   999     CONTINUE
04500           RETURN
04600           END
```

```
00100   C
00200           SUBROUTINE GMAIN(NPAR,PARVAL,ITMAX,PAREPS,FNEPS,INPOL,
00300         1     PLM,PLMAX,PLMULT,PLDIVD,STPMAX)
00400   C
00500   C  Gradient search minimisation program based on a modified
00600   C  Newton method. The modifications are based on the ideas
00700   C  of Marquardt( see Bard 1974) and Goldfeld,Quandt and Trotter
00800   C  (1966). The search is restricted to the space of dominant
00900   C  eigenvalues, to prevent long steps in the directions of
01000   C  non-identifyable parameters. All eigenvalues of the normalised
01100   C  information matrix(or hessian) that are less than the value
01200   C  EPS times the largest eigenvalue are set to zero so that a
01300   C  pseudo-inverse of the information matrix is used( see sub-
01400   C  routine GSTEP).At the time of writing EPS=0.000001.
01500   C
01600   C  A variable value PLM is added to each non zero modified
01700   C  eigenvalue to reduce the size of the search steps if the
01800   C  logic of the program decides that the assumption of a quad-
01900   C  ratic response surface is not satisfactory.If the search
02000   C  reaches a region where the modified Newton steps are being
02100   C  rejected, or the predicted function value is quite different
02200   C  from the function value actually obtained, the assumption of
02300   C  quadratitity is deemed poor and the parameter PLM is increased
02400   C  by multiplying by the factor PLMULT. If the reverse is found
02500   C  to be true, then PLM is divided by the factor PLDIVD.  The
02600   C  initial value of PLM (at the time of writing is set at 10**-7.
02700   C  Hence the first step is essentially a newton step.
02800   C
02900   C  If a step is found to be acceptable( function at the new point
03000   C  FNEW is smaller than the function value at the previous best
03100   C  point FZERO ) then one extrapolation step is attempted.  If
03200   C  FNEW > than or equal to FZERO then a number of interpolation
03300   C  steps are tried, until an improvement of the function value
03400   C  is observed or one of the convergence criteria is met.
03500   C  Two options are available for extrapolation or interpolation.
03600   C  If user specifies INPOL = 1 then the current step is halved,
03700   C  or doubled as appropriate.  If INPOL = 2 then a quadratic
03800   C  interpolation or extrapolation step is attempted. the maximum
03900   C  extrapolation step factor is set to STPMAX (=3 at time of
04000   C  writing).
04100   C
04200   C  There are three ways in which normal termination of the search
04300   C  can occour.The user specifies a) max no. of iterations
04400   C  b) a parameter stepsize convergence criterion that must be
04500   C  met by all the parameters  c) a function value convergence
04600   C  criterion. All three must be specified by the user.
04700   C
04800   C
04900   C   The following info must be specified by the user through the
05000   C  call statement:
05100   C                 PARVAL(i)- initial parameter values
05200   C                 NPAR     - number of parameters
05300   C                 ITMAX    - max no. of iterations
05400   C                 PAREPS   - minimum fractional change of the par-
05500   C                            ameter values
05600   C                 FNEPS    - minimum fractional change of the
05700   C                            function value
05800   C                 INPOL    - option on type of interpolation/extrap
05900   C                            oletion step.
06000   C                                 =1  halve/double step
06100   C                                 =2  quadratic int/ext step
06200   C
06300   C  The user must also provide a routine called MFUNC which must
06400   C  provide the fn. value,gradient vector SLOP(i) and the information
06500   C  matrix( this may be the Hessian or an approximation to it)
06600   C  H(i,j).  A parameter INFO is sent to MFUNC through the call
06700   C  statement so that when INFO=1 all information should be
06800   C  computed(fn value,gradient and information matrix). When
06900   C  INFO=2 only function value information should be computed.
07000   C  The format of the subroutine statement is;
07100   C                 SUBROUTINE MFUNC(NPAR,PARVAL,FZERO,SLOP,H,INFO)
07200   C
```

```
07300  C-----------------------------------------------------------------
07400  C
07500  C  Explanation of program variables
07600  C
07700  C        FZERO   - function value at current best point
07800  C        FNEW    - function value at the point being tested
07900  C        FHAT    - predicted value of the function(quadratic approx)
08000  C        FTEMP   - stores intermediate function value while an
08100  C                  extrapolation step is being tried.
08200  C        THETA(i)- stores best parameter values so far obtained
08300  C        TSLOP(i)- stores gradient at best par vals
08400  C        TH(i,j) - stores info matrix at best par vals
08500  C        PARVAL(i)- parameter values sent to the model(s)
08600  C        SLOP(i)  - gradient at parval(i)
08700  C        H(i,j)   - info matrix at parval(i)
08800  C        DELPAR(i)- change in the par values predicted by the
08900  C                   constrained quadratic approximation by
09000  C                   subroutine GSTEP
09100  C        INFO    - tells function evaluating subroutin whether to
09200  C                  compute the gradient and information matrix
09300  C        PLM     - levenberg-marquardt parameter
09400  C        PLMAX   - max permissible value of PLM
09500  C        PLMULT  - factor to increase PLM by if quad approx poor
09600  C        PLDIVD  - factor to divide PLM by if qued approx is good
09700  C        STEP    - stepsize factor - normally equal to 1
09800  C        STPMAX  - max permissible stepsize factor
09900  C        KMOD1   - no of fn calls with INFO=1
10000  C        KMOD2   - no of fn calls with INFO=2
10100  C        K       - total no of function calls
10200  C        C       - gradient in the direction of the search vector
10300  C        INPOL   - option parameter for extrapolation/interpolation
10400  C
10500  C  The following subroutines are called by this program
10600  C
10700  C        MFUNC(NPAR,PARVAL,FZERO,SLOP,H,INFO)
10800  C                -function,gradient,information matrix evaluator
10900  C        GSTEP(NPAR,H,SLOP,DELPAR,PLM)
11000  C                -computes step to be tried
11100  C
11200  C
11300  C=================================================================
11400  C
11500           DOUBLE PRECISION SLOP(10),H(10,10),DELPAR(10)
11600           DIMENSION PARVAL(10)
11700           DIMENSION THETA(10),TSLOP(10),TH(10,10)
11800  C
11900  C=================================================================
12000  C
12100  C
12200  C-----------------------------------------------------------------
12300  C    Initialise values
12400  C
12500           KMOD1=1
12600           KMOD2=0
12700           INFO=1
12800  C
12900  C---------------------- ------------------------------------------
13000   444    CONTINUE
13100           INFO=1
13200           WRITE(5,50)
13300   50      FORMAT(5X,50('-'),/)
13400  C
13500  C     Call the function to be minimised to get function value
13600  C        and the values of the partial derivatives (INFO=1)
13700  C        at the starting parameter values.
13800  C
13900           CALL MFUNC(NPAR,PARVAL,FZERO,SLOP,H,INFO)
14000  C
```

```
C-------------------------------------------------------------------
C  The program loops back to here when a successful step and
C  extension or a sucessful interpolation step is obtained.
C
 555     CONTINUE
         INFO=1
C    Store initial parameter values
C
         FTEMP=FZERO
         DO 115 I=1,NPAR
         THETA(I)=PARVAL(I)
         TSLOP(I)=SLOP(I)
         DO 114 J=1,NPAR
         TH(I,J)=H(I,J)
 114     CONTINUE
 115     CONTINUE
C
         K=KMOD1+KMOD2
C
C  Print out the summary of the present stage of the search
C
         WRITE(5,125)K,KMOD1,PLM
 125     FORMAT(5X,'Iter no',I4,5X,'No successful ',I4,
     1   5X,'Lev-Marq par ',G12.6,/)
         WRITE(5,200)(PARVAL(I),I=1,NPAR)
 200     FORMAT(3X,5(G12.6,X),/,5X,5(G12.6,X),/)
         IF(INFO.EQ.1)WRITE(5,221)(SLOP(I),I=1,NPAR)
 221     FORMAT(3X,'SLOP',5(G12.6,X),/,5X,5(G12.6,X),/)
         WRITE(5,211)FZERO
 211     FORMAT(5X,'Function value(FZERO) = ',G16.10,/,50('-'),/)
C
C    Compute the optimal constrained quadratic step DELPAR(i)
C
         CALL GNSTEP(NPAR,H,SLOP,DELPAR,PLM)
C
C  Compute the quadratic prediction of the function value FHAT
C      -
         C=0.0
         DO 276 I=1,NPAR
         C=C+SLOP(I)*DELPAR(I)
 276     CONTINUE
         FHAT=FZERO+0.5*C
C
C??????????????????????????
C  Temporary check
         IF(C.LE.0.0)GO TO 277
         WRITE(5,278)C,FHAT
 278     FORMAT(5X,'******* C = ',F20.10,'  FHAT=  ',F20.10)
         STOP
 277     CONTINUE
C??????????????????????????
C
C    Initially set step augumentation par = 1
         STEP=1.0
C
C  Program loops here when stepsize parameter is changed
C
 777     CONTINUE
C
 96      CONTINUE
         DO 100 I=1,NPAR
         PARVAL(I)=THETA(I)+STEP*DELPAR(I)
C
         IF( PARVAL(I).GE.0.0 ) GO TO 98
         IF(STEP.GT.1.0) GO TO 97
         STEP=STEP/2.0
         GO TO 96
 97      CONTINUE
         DEL=STEP-1.0
         STEP=1.0+DEL/2.0
         GO TO 96
C
 98      CONTINUE
C
 100     CONTINUE
```

```
C
C     Obtain function value at the new parameter values
C
          CALL MFUNC(NPAR,PARVAL,FNEW,SLOP,M,INFO)
          IF(INFO.EQ.1)KMOD1=KMOD1+1
          IF(INFO.EQ.2)KMOD2=KMOD2+1
          K=KMOD1+KMOD2
C
C  Print out the summary of the present stage of the search
C
          IF(INFO.EQ.1)WRITE(5,125)K,KMOD1,PLM
          IF(INFO.EQ.2)WRITE(5,126)K,KMOD1
          WRITE(5,200)(PARVAL(I),I=1,NPAR)
  126     FORMAT(5X,'Iteration no. ',I4,5X,'No successful ',I4,/)
          IF(INFO.EQ.1)WRITE(5,223)(SLOP(I),I=1,NPAR)
  223     FORMAT(3X,'SLOP',5(G12.6,X),/,5X,5(G12.6,X),/)
          WRITE(5,212)FNEW
  212     FORMAT(5X,'Function value(FNEW) = ',G16.10,/,50('-'),/)
C
C    Check iteration convergence
          IF(K.GE.ITMAX) GO TO 999
C
C-----------------------------------------------------------------
C  Evaluate the step and decide how to proceed
c
C     Check for parameter value change convergence
          I=1
  499     CONTINUE
          A=ABS(STEP*DELPAR(I)/THETA(I))
          IF(A.GT.PAREPS)GO TO 501
          IF(I.GE.NPAR)GO TO 998
          I=I+1
          GO TO 499
C
  501     CONTINUE
C  At least one parameter has not converged so continue search
C
C  Check if the new function value is acceptable
C
          IF(FNEW.LT.FTEMP)GO TO 502
C
C  The step is unacceptable, reject the step.
C
          WRITE(5,702)
  702     FORMAT(5X,'Step rejected')
C
C  If the last step was not an extrapolation, an interpolation
C  to reduce the step size may be tried.
C  If STEP greater than one then last step was an extrapolation
          IF(STEP.LE.1)GO TO 503
C
C  Reject the extrapolation step
          DO 504 I=1,NPAR
          PARVAL(I)=THETA(I)+DELPAR(I)
  504     CONTINUE
          FZERO =FTEMP
          GO TO 555
C
  503     CONTINUE
C  Try an interpolation step, last step was not an extrapolation
C  If INPOL is equal to 1, then halve step,  else compute a
C  quadratic interpolation step.
c
          IF(INPOL.EQ.2)GO TO 506
          WRITE(5,707)
  707     FORMAT(5X,'Step being halved')
          STEP=STEP/2.0
          GO TO 507
  506     CONTINUE
          WRITE(5,708)
  708     FORMAT(5X,'Step reduced thru quadratic interpolation')
```

```
C??????????????????
C       C=0.0
C       DO 508 I=1,NPAR
C       C=C+SLOP(I)*DELPAR(I)
C 508   CONTINUE
C???????????????????
        STEP=0.5*(C*STEP**2)/(C*STEP-FZERO-FNEW)
 507    CONTINUE
        INFO=2
        PLM=PLM*PLMULT
        IF(PLM.GT.PLMAX)PLM=PLMAX
        GO TO 777
C
C-------------------------------------------------------------
C
 502    CONTINUE
C  The function value was reduced so step is accepted
C
        FTEMP=FNEW
C
        WRITE(5,601)
 601    FORMAT(5X,'Step accepted')
C
C  Test for function convergence
C
        A=ABS( (FZERO-FNEW)/FZERO )
        IF(A.GT.FNEPS)GO TO 511
        WRITE(5,722)
 722    FORMAT(5X,'Termination by function convergence')
        GO TO 997
C
 511    CONTINUE
C  Extrapolate only if there were no previous extrapolations
C  or interpolation in this trial.
C
        IF(STEP.EQ.1)GO TO 513
        FZERO=FTEMP
        GO TO 444
C
 513    CONTINUE
C  Check if the quadratic fit is acceptable
        Z=(FZERO-FNEW)/(FZERO-FMAT)
        IF(Z.GE.0.7.AND.Z.LE.1.3)GO TO 514
        WRITE(5,716)
 716    FORMAT(5X,'Quadratic fit not acceptable, no extrapolation')
        PLM=PLM/PLDIVD
        IF(PLM.LT.0.0000001)PLM=0.0000001
        FZERO=FTEMP
        GO TO 555
C
 514    CONTINUE
        WRITE(5,717)
 717    FORMAT(5X,'Quadratic fit acceptable, try extrapolation')
C
        IF(INPOL.EQ.2)GO TO 516
        WRITE(5,719)
 719    FORMAT(5X,'Step being doubled')
        STEP=STEP*2.0
        INFO=2
        IF(PLM.GT.0.0000001)PLM=PLM/PLDIVD
        GO TO 777
 516    CONTINUE
        WRITE(5,721)
 721    FORMAT(5X,'Quadratic extrapolation step')
C???????????????????
C       C=0.0
C       DO 518 I=1,NPAR
C       C=C+SLOP(I)*DELPAR(I)
C 518   CONTINUE
C???????????????????
```

```
34300           STEP=0.5*(C*STEP**2)/(C*STEP-FZERO-FNEW)
34400           IF(STEP.GT.1.1)GO TO 521
34500           INFO=1
34600           IF(PLM.GT.0.0000001)PLM=PLM/PLDIVD
34700           WRITE(5,713)
34800    713    FORMAT(5X,'Step less than 1.1 so no extrapolation')
34900           FZERO=FTEMP
35000           GO TO 555
35100   C
35200    521    CONTINUE
35300           INFO=2
35400           IF(PLM.GT.0.0000001)PLM=PLM/PLDIVD
35500           WRITE(5,723)STEP
35600    723    FORMAT(5X,'Quadratic step extrap. par.=',F10.6)
35700           IF(STEP.LE.STPMAX)GO TO 777
35800           STEP=STPMAX
35900           WRITE(5,724)
36000    724    FORMAT(5X,'Step set equal to stepmax')
36100           GO TO 777
36200   C
36300   C-----------------------------------------------------------------
36400   C
36500   C  Termination of the search
36600   C
36700    997    CONTINUE
36800           WRITE(5,802)
36900    802    FORMAT(5X,'*** Termination due to function convErgence')
37000           GO TO 990
37100    998    CONTINUE
37200           WRITE(5,803)
37300    803    FORMAT(5X,'*** Termination due to parameter convergence')
37400           GO TO 990
37500    999    CONTINUE
37600           WRITE(5,804)
37700    804    FORMAT(5X,'*** Termination due to iter.limit exceedance')
37800    990    CONTINUE
37900   C
38000   C  Print final values of the parameters, gradient vector and
38100   C  the information matrix.
38200   C
38300           WRITE(5,851)(THETA(I),I=1,NPAR)
38400    851    FORMAT(3X,'Final parameters',/,3X,5(G12.6,X),/,5X,
38500          1  5(G12.6,X),/)
38600           WRITE(5,852)(TSLOP(I),I=1,NPAR)
38700    852    FORMAT(3X,'Gradients',/,3X,5(G12.6,X),/,5X,5(G12.6,
38800          1  X),/)
38900           DO 853 I=1,NPAR
39000           WRITE(5,854)(TH(I,J),J=1,I)
39100    854    FORMAT(3X,'Info',X,10(G12.6,X) )
39200    853    CONTINUE
39300   C
39400   C
39500   C
39600   C=================================================================
39700   C  Last lines of program follow
39800           RETURN
39900           END
40000   C
40100   C
```

```
C-----------------------------------------------------------------
      SUBROUTINE GNSTEP(NPAR,H,SLOP,DELPAR,PLM)
C     ----------
C
C  Note : no normalisation
C
C-----------------------------------------------------------------
C  Computes a Levenberg-Marquardt step in the parameter space.
C  Part of a Gradient algorithm to minimise a function of ten
C  or less parameters using successive quadratic approximations.
C
C  The Levenberg-Marquardt par. PLM is specified through the
C  calling program.  If PLM=0.0, a Newton step in the subspace
C  of dominant eigenvalues results.  By setting PLM not equal to
C  zero (positive values only) we constrain the size of the
C  step in an attempt to restrict it to a region in which the
C  quadratic approximation is more valid.
C
C  Also specified through the calling program are the following.
C      NPAR    - no. of parameters
C      H(i,j)  - information matrix
C      SLOP(i) - matrix of first partial derivatives wrt. pars.
C
C  The following information is returned to the calling program
C      DELPAR(i) - recommended change in the par values
C                  to get a reduction in the function.
C
C-----------------------------------------------------------------
C
      DOUBLE PRECISION H(10,10),U(10,10)
      DOUBLE PRECISION HEIG(10),B(10,10),SLOP(10)
      DOUBLE PRECISION EPS,EPS2,AKO,DUM1(10),CUTOFF,PLM1
      DOUBLE PRECISION DELPAR(10),A
C
      DATA EPS/0.00000001/,EPS2/0.00000001/,AKO/1.0D3B/
C
C-----------------------------------------------------------------
C
C
C
C
      DO 110 I=1,NPAR
      DO 120 J=1,NPAR
      U(I,J)=H(I,J)
  120 CONTINUE
  110 CONTINUE
C
C  ***  Compute the Eigenvalues and Eigenvectors of H
C
C
      NT=NPAR
      CALL SYMGRD(U,HEIG,DUM1,AKO,NT,10,EPS2,.FALSE.,.TRUE.,
     1   .FALSE.,IFAIL)
C
      CALL DSORT(NT,HEIG,U)
C
      DO 998 I=1,NPAR
      WRITE(5,997)HEIG(I)
  997 FORMAT(3X,'**** LAMDA  = ',3X,G20.6)
  998 CONTINUE
C
C  ***  Compute the (PSEUDO?) inverse of the Eigenvector matrix
C
C
C
C  ***  Count no. of eigvals larger than EPS*(max HEIG(I))
C
C
      NEIG=NPAR
      CUTOFF=EPS*( HEIG(1) )
      DO 150 I=2,NPAR
      IF(HEIG(I).GT.CUTOFF) GO TO 150
      NEIG=I-1
      GO TO 160
  150 CONTINUE
  160 CONTINUE
C
```

```
C  *** Compute a Levenberg-Marquardt(?) step with given PLM
C
C  *** Compute the trial step DELPAR
C
      WRITE(5,99)NPAR,NEIG
 99   FORMAT(5X,'NPAR=',I2,2X,'NEIG=',I2)
C
      DO 170 I=1,NEIG
      A=0.0
      DO 180 J=1,NPAR
      A=A+U(J,I)*SLOP(J)
 180  CONTINUE
      DO 190 J=1,NPAR
      B(J,I)=A*U(J,I)
 190  CONTINUE
 170  CONTINUE
C
C
      PLM1=PLM*HEIG(1)
      DO 200 I=1,NPAR
      DELPAR(I)=0.0
      DO 210 J=1,NEIG
      DELPAR(I)=DELPAR(I)-( B(I,J)/(HEIG(J)+PLM1) )
 210  CONTINUE
 200  CONTINUE
C
C
      WRITE(5,899)
899   FORMAT(3X,'DELPAR')
      WRITE(5,900)(DELPAR(I),I=1,NPAR)
 900  FORMAT(3X,5(G12.6,1X),/,5X,5(G12.6,1X))
C-----------------------------------------------------------
C     WRITE(5,898)
C898  FORMAT(3X,'HESSIAN')
C     DO 910 I=1,NPAR
C     WRITE(5,920)(H(I,J),J=1,NPAR)
C 920 FORMAT(3X,5(G12.6,1X),/,5X,5(G12.6,1X))
C-----------------------------------------------------------
 910  CONTINUE
      RETURN
C  *** Last line of subroutine follows
      END
```

```
00100   C
00200           SUBROUTINE MFUNC(NPAR,PARVAL,FUNC,SLOP,H,INFO)
00300   C
00400   C-----------------------------------------------------------------
00500   C
00600   C   ***  COMPUTES FUNCTION VALUE AND THE PARTIAL DERIVATIVES WRT
00700   C   ***  PARAMETERS IF DESIRED.
00800   C
00900   C-----------------------------------------------------------------
01000   C
01100           DOUBLE PRECISION SLOP(10),H(10,10)
01200   C
01210           COMMON/CNTROL/ISERCH,IOP,ISENSE,IMODEL,IREPAR
01215           COMMON/ESTIM/PARA(10),GRADS(10),FISHER(10,10)
01220   C
01300           DIMENSION PARVAL(10)
01400   C
01500   C-----------------------------------------------------------------
01600   C
01700   C   ***  INVERT THE PARAMETERIZATION FROM PARVAL(I) TO VALUES OF
01800   C   ***  THE MODEL PARAMETERS.
01900   C
02000           CALL PARIZ2(PARVAL)
02100   C
02200   C-----------------------------------------------------------------
02300   C
02400           IF(INFO.EQ.1)GO TO 100
02500   C
02600   C-----------------------------
02700   C
02800   C   ***  COMPUTE ONLY FUNCTION VALUE
02900   C
03000           IF(IMODEL.EQ.1)CALL MODLZ1
03100   C
03200           IF(IMODEL.EQ.2)CALL MODLA1
03300   C
03400           CALL LOGLIK(FUNC)
03500   C
03600           RETURN
03700   C
03800   C-----------------------------
03900   C
04000   100     CONTINUE
04100   C
04200   C   ***  COMPUTE FUNCTION VALUE AND SLOP AND INFORMATION MATRICES
04300   C
04400           IF(IMODEL.EQ.1)CALL MODELZ
04500   C
04600           IF(IMODEL.EQ.2)CALL MODELA
04700   C
04710           IF(IREPAR.NE.1)GO TO 500
04720           CALL PARIZ3
04730   500     CONTINUE
04740   C
04800           CALL LOGLIK(FUNC)
04900   C
05000           CALL DERIV1
05100   C
05200           CALL DERIV2
05300   C
05400   C-----------------------------------------------------------------
05500   C
05550   C   ***  CONVERT FROM SINGLE PRECISION TO DOUBLE PRECISION
05575   C
05600           DO 200 I=1,NPAR
05700           SLOP(I)=GRADS(I)
05800           DO 190 J=1,NPAR
05900           H(I,J)=FISHER(I,J)
06000   190     CONTINUE
06100   200     CONTINUE
06200   C
06300   C
06400   C-----------------------------------------------------------------
06500   C
06600           RETURN
06700           END
```

```
C-ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
C
C   ***   CONTAINS SUBROUTINES THAT COMPUTE THE PARTIAL
C   ***   DERIVATIVES OF THE FUNCTION WRT THE PARAMETERS.
C
C-----------------------------------------------------------------------
C
          SUBROUTINE DERIV1
C
C-----------------------------------------------------------------------
C
          COMMON/CNTROL/ISERCH,IOP,ISENSE,IMODEL,IREPAR
          COMMON/RINPUT/NDATA,P(999),OBSG(999),TRUQ(999)
          COMMON/ROUTPT/G(999),R(999),S(999),B(999),PPERC(999),PERC(999)
          COMMON/ESTIM2/DG(999,6)
          COMMON/ESTIM/PARVAL(10),SLOP(10),HESS(10,10)
          COMMON/PARAMS/NPAR,UM,UK,BM,BK,Z,X,USI,BSI
          COMMON/PARAM2/GAMA,PHI1,SIGSGR
          COMMON/ERROR/V(999),EPS(999)
          COMMON/DERROR/DV(999,10),DVGAMA(999),DEPS(999,10)
C
C-----------------------------------------------------------------------
C
          IF(IOP.EQ.2)GO TO 300
C
C-----------------------------------------------------------------------
C
C   ***   COMPUTE THE GRADIENTS OF THE SLS FUNCTION WRT PARAMETERS
C
          DO 100 K=1,NPAR
C
          SUM=0.0
          DO 200 I=1,NDATA
          SUM=SUM + ( G(I)-OBSG(I) )*DG(I,K)*2.0
  200     CONTINUE
C
          SLOP(K)=SUM
C
  100     CONTINUE
C
          GO TO 999
C
C-----------------------------------------------------------------------
C
  300     CONTINUE
C
C   ***   DERIVATIVES OF THE NLL FUNCTION BASED ON THE RATING
C   ***   CURVE MODEL
C
          NTHETA = 6
C
          DO 400 I=1,NDATA
          DO 390 J=1,NTHETA
          DV(I,J)= -DG(I,J)*( G(I)**(GAMA-1.0) )
  390     CONTINUE
          DUM1=( OBSG(I)**GAMA )*(GAMA*ALOG(OBSG(I))-1.0)
          DUM2=( G(I)**GAMA )*(GAMA*ALOG(G(I))-1.0)
          DVGAMA(I)= ( DUM1 - DUM2 ) / ( GAMA*GAMA )
  400     CONTINUE
```

```
C
      SUMLOG=0.0
      DO 500 I=2,NDATA
      SUMLOG=SUMLOG + ALOG( OBSG(I) )
      DO 490 J=1,NTHETA
      DEPS(I,J) = DV(I,J) - PHI1*DV(I-1,J)
490   CONTINUE
      DEPS(I,NTHETA+1) = -V(I-1)
      DEPS(I,NTHETA+2) = DVGAMA(I) - PHI1*DVGAMA(I-1)
500   CONTINUE
C
      IF(IREPAR EQ 1)CALL PAR124
C
C  **   COMPUTE THE GRADIENTS
C
      DO 600 K=1,NPAR
      SUM=0.0
      DO 590 I=2,NDATA
      SUM=SUM + EPS(I)*DEPS(I,K)
590   CONTINUE
      SLOP(K) = SUM/SIGSQR
600   CONTINUE
C*****************************
      SLOP(NPAR)=SLOP(NPAR) - SUMLOG
C     SLOP(NPAR)=SLOP(NPAR)  <--------
C-------------------------------------------------------------
C
999   CONTINUE
C
      RETURN
      END
```

```
C ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
C
      SUBROUTINE DERIV2
C
C-----------------------------------------------------------------
C
C  ***  CALCULATES THE FISHER INFORMATION MATRIX USING
C  ***  GIVEN SENSITIVITY COEFFICIENTS DQ(NDATA,NPAR).
C
C-----------------------------------------------------------------
C
      COMMON/ESTIM2/DQ(999,6)
      COMMON/ESTIM/PARVAL(10),SLOP(10),HESS(10,10)
      COMMON/PARAMS/NPAR,UM,UK,BM,BK,Z,X,USI,BSI
      COMMON/RINPUT/NDATA,P(999),QBSQ(999),TRUQ(999)
      COMMON/DERROR/DV(999,10),DVGAMA(999),DEPS(999,10)
      COMMON/PARAM2/GAMA,PHI1,SIGSQR
      COMMON/CNTROL/ISERCH,IOP,ISENSE,IMODEL,IREPAR
C
C-----------------------------------------------------------------
C
      DO 10 I=1,NPAR
      DO 15 J=1,NPAR
      HESS(I,J)=0.0
15    CONTINUE
10    CONTINUE
C
C---------------------------------------------------------------
C
      IF(IOP.EQ.2)GO TO 500
C
C---------------------------------------------------------------
C
C  *** HESSIAN MATRIX FOR THE SLS FUNCTION
C
      DO 20 I=1,NDATA
C
      DO 30 K=1,NPAR
      DO 25 J=1,NPAR
      HESS(K,J)=HESS(K,J) + 2.0*DQ(I,K)*DQ(I,J)
25    CONTINUE
30    CONTINUE
C
20    CONTINUE
      GO TO 999
C
C-----------------------------------------------------------------
C
500   CONTINUE
C
C  ***  HESSIAN OF THE NLL FUNCTION BASED ON THE RATING CURVE
C  ***  MODEL ( APPROXIMATION APPROACH # 1 ).
C
      DO 220 I=2,NDATA
      DO 230 K=1,NPAR
      DO 225 J=1,NPAR
      HESS(K,J) = HESS(K,J) + DEPS(I,K)*DEPS(I,J)/SIGSQR
225   CONTINUE
230   CONTINUE
220   CONTINUE
C
C-----------------------------------------------------------------
C
999   CONTINUE
C
      RETURN
      END
```

```
C--ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
C
C  ***  CONTAINS THE REPARAMETERIZATION SUBROUTINES
C
C---------------------------------------------------------------------
CZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
C
      SUBROUTINE PARIZ1
C
C---------------------------------------------------------------------
C
      COMMON/PARAMS/NPAR,UM,UK,BM,BK,Z,X,USI,BSI
      COMMON/REPAR/A,RKU,RKB,XINV,AINV
      COMMON/PARAM2/GAMA,PHI1,SIGSQR
C
      COMMON/ESTIM/PARVAL(10),SLOP(10),HESS(10,10)
C
C---------------------------------------------------------------------
C
      A = Z
C     RKU=1.0/UK
C     RKB=1.0/BK
C     XINV=1.0/X
C     AINV=1.0/A
C     XINV=ALOG(X+1.0)
C     RA=A**(1.0/X)
C
      PARVAL(1)=UM
      PARVAL(2)=UK
      PARVAL(3)=BM
      PARVAL(4)=BK
      PARVAL(5)=GAMA
      PARVAL(6)=X
      PARVAL(7)=PHI1
      PARVAL(8)=A
C
C--------------------------------------------------------------------
C
      RETURN
      END
C
```

```
CZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
C
      SUBROUTINE PARI22(PARVAL)
C
C-----------------------------------------------------------------
C
      COMMON/PARAMS/NPAR,UM,UK,BM,BK,Z,X,USI,BSI
      COMMON/PARAM2/GAMA,PHI1,SIGSQR
      COMMON/REPAR/A,RKU,RKB,XINV,AINV
C
C-----------------------------------------------------------------
C
      DIMENSION PARVAL(10)
C
C-----------------------------------------------------------------
C
      UM=PARVAL(1)
      UK=PARVAL(2)
      BM=PARVAL(3)
      BK=PARVAL(4)
      GAMA=PARVAL(5)
      X=PARVAL(6)
      PHI1=PARVAL(7)
      A=PARVAL(8)
C
C     UK=1.0/RKU
C     BK=1.0/RKB
C     X=1.0/XINV
C     A=1.0/AINV
C     X=EXP(XINV)-1.0
C     A=0.27**X
      Z=A
C     IF(A.GT.0.9)A=0.9
C
C-----------------------------------------------------------------
C
      RETURN
      END
C
C=================================================================
```

```
C
CZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
C
      SUBROUTINE PARIZ3
C
C-----------------------------------------------------------------------
C
C  ***  REPARAMETERIZES THE DERIVATIVES OF THE FLOWS WRT. THE
C  ***  PARAMETERS IF THE STUDY IS TO BE CONDUCTED IN A
C  ***  REPARAMETERIZED PARAMETER SPACE.
C
C-----------------------------------------------------------------------
C
      COMMON/RINPUT/NDATA,P(999),OBSQ(999),TRUQ(999)
      COMMON/PARAMS/NPAR,UM,UK,BM,BK,Z,X,USI,BSI
      COMMON/PARAM2/GAMA,PHI1,SIGSQR
      COMMON/REPAR/A,RKU,RKB,XINV,AINV
      COMMON/ESTIM2/DQ(999,6)
C
C-----------------------------------------------------------------------
C
C
C     DO 100 I=1,NDATA
C     DQ(I,2)=DQ(I,2)*( -UK*UK )
C     DQ(I,4)=DQ(I,4)*( -BK*BK )
C     DQ(I,6)=DQ(I,6)*( -X*X )
C     DQ(I,6)=DQ(I,6) * (X+1)
C
C     DUMA=DQ(I,5)
C     DQ(I,5)=DQ(I,5)*X*A**( (X-1)/X )
C     DQ(I,6)=DQ(I,6) + DUMA*ALOG(A)*A/X
C
C     DUM=DQ(I,1)
C     DQ(I,1)=DQ(I,7)
C     DQ(I,7)=DUM
C
C     DUM=DQ(I,2)
C     DQ(I,2)=DQ(I,8)
C     DQ(I,8)=DUM
C
  100 CONTINUE
C
C
      RETURN
      END
C
```

```
C
      SUBROUTINE PARIZ4
C
C-----------------------------------------------------------------------
C
      COMMON/RINPUT/NDATA,P(999),OBSQ(999),TRUQ(999)
      COMMON/DERROR/DV(999,10),DVGAMA(999),DEPS(999,10)
C
C-----------------------------------------------------------------------
C
      DO 100 I=1,NDATA
      DUM1=DEPS(I,8)
      DEPS(I,8)=DEPS(I,5)
      DEPS(I,5)=DUM1
  100 CONTINUE
C
C-----------------------------------------------------------------------
C
      RETURN
      END
```

```
C-ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
C
      SUBROUTINE SIMPLX(NPAR,ITMAX,PAREPS,FNEPS,A)
C
C-----------------------------------------------------------------
C
      DIMENSION A(10),P(10,11),Y(11),PBAR(10),PSTAR(10),PPSTAR(10
C
C-----------------------------------------------------------------
C
      DATA RCOEFF/1.0/,ECOEFF/2.0/,CCOEFF/0.5/
      DATA BIGNUM/10000000000.0/,KONVGE/5/,STEP/0.05/
C
C-----------------------------------------------------------------
C
      WRITE(5,400)
400   FORMAT(/10('*'),'SIMPLEX DIRECT SEARCH'10('*'),/,
     1  10X,'NONLINEAR OPTIMIZATION ALGORITHM',/)
C
C-----------------------------------------------------------------
C
      N=NPAR
      N1=N+1
      RN=FLOAT(N)
C
C-----------------------------------------------------------------
C
C  PRINT OUT INITIAL PARAMETER VALUES
C
      WRITE(5,203)
203   FORMAT(10X,'INITIAL PARAMETER VALUES',/)
C
      DO 214 I=1,N
C
      WRITE(5,204)I,A(I)
204   FORMAT(3X,I5,4X,G20.7)
C
214   CONTINUE
C
C-----------------------------------------------------------------
C
C  CONSTRUCT THE INITIAL SIMPLEX
C
1001  CONTINUE
      DO 1 I=1,10
      P(I,N1)=A(I)
1     CONTINUE
C
      CALL SXFUNC(N,A,FVALUE)
      Y(N1)=FVALUE
      ICOUNT=ICOUNT+1
C
      WRITE(5,700)
700   FORMAT(/,10X,'INITIAL SIMPLEX')
C
      WRITE(5,701)ICOUNT,Y(N1),(P(I,N1),I=1,N)
701   FORMAT(3X,I5,X,'FUNC=',G12.5,'PARS  ',10(G12.5,X))
C
      DO 2 J=1,N
      DCHK=A(J)
      A(J)=A(J)+STEP*A(J)
      DO 3 I=1,10
      P(I,J)=A(I)
3     CONTINUE
      CALL SXFUNC(N,A,FVALUE)
      Y(J)=FVALUE
      ICOUNT=ICOUNT+1
C
      WRITE(5,702)ICOUNT,Y(J),(P(I,J),I=1,N)
702   FORMAT(3X,I5,X,'FUNC=',G12.5,'PARS  ',10(G12.5,X))
C
      A(J)=DCHK
2     CONTINUE
C
      WRITE(5,600)
600   FORMAT(10X,'SIMPLEX CONSTRUCTION COMPLETE',/)
C
C-----------------------------------------------------------------
```

```
C
C  CHECK ON NUMBER OF ITERATIONS
C
1002    CONTINUE
C
        IF(ICOUNT.LT.ITMAX)GO TO 1003
        KIST=3
        GO TO 900
C
C-----------------------------------------------------------
C
C  FIND HIGH, SECOND HIGH, AND LOW POINTS
C
1003    CONTINUE
C
        YLO=Y(1)
        ILO=1
        DO 51 I=2,N1
        IF(Y(I).GE.YLO)GO TO 52
        YLO=Y(I)
        ILO=I
52      CONTINUE
51      CONTINUE
C
        YSECHI=YLO
        YHIGH=YLO
        ISEC=ILO
        IHI=ILO
        DO 41 I=1,N1
        IF(Y(I).GT.YHIGH)GO TO 5
        IF(Y(I).GT.YSECHI)GO TO 6
        IF(Y(I).LT.YLO)GO TO 7
        GO TO 4
C
5       CONTINUE
        YSECHI=YHIGH
        ISEC=IHI
        YHIGH=Y(I)
        IHI=I
        GO TO 4
C
6       CONTINUE
        YSECHI=Y(I)
        ISEC=I
        GO TO 4
C
7       CONTINUE
C
        YLO=Y(I)
        ILO=I
C
4       CONTINUE
C
41      CONTINUE
C
        WRITE(5,727)YHIGH,YSECHI,YLO
727     FORMAT(/,3X,'HI=',G14.6,' SEC=',G14.6,' LOW=',G14.6)
C
C-----------------------------------------------------------
C
C  CONVERGENCE CHECK ON THE FUNCTION VALUE
C
1004    CONTINUE
C
        DCHK=ABS((YHIGH-YLO)/YHIGH)
        WRITE(5,61)DCHK
61      FORMAT(3X,'FUNCTION CONVG CHECK ',G12.6)
        IF(DCHK.GE.FNEPS)GO TO 1005
        KIST=1
        GO TO 900
C
C-----------------------------------------------------------
C
C  PARAMETER COVERGENCE CHECK
C  DONE EVERY 5 SIMPLEXES
C
1005    CONTINUE
C
```

```
15700           KONVGE=KONVGE-1
15800           IF(KONVGE.NE.0)GO TO 1012
15900           KONVGE=5
16000           WRITE(5,787)
16100   787     FORMAT(3X,'PAR CONVG CHECK')
16200   C
16300           DO 10 I=1,N
16400           COORD1=P(I,1)
16500           COORD2=COORD1
16600   C
16700           DO 11 J=2,N1
16800           IF(P(I,J).GE.COORD1)GO TO 12
16900           COORD1=P(I,J)
17000   C
17100   12      CONTINUE
17200           IF(P(I,J).LE.COORD2)GO TO 11
17300           COORD2=P(I,J)
17400   C
17500   11      CONTINUE
17600           DCHK=ABS( (COORD2-COORD1)/COORD2 )
17700           WRITE(5,62)DCHK
17800   62      FORMAT(3X,'PAR CHECK ',G14.6)
17900           IF(DCHK.GT.PAREPS)GO TO 1012
18000   C
18100   10      CONTINUE
18200           KIST=2
18300           GO TO 900
18400   C
18500   C-----------------------------------------------------------
18600   C
18700   C  COMPUTATION OF PBAR
18800   C
18900   1012    CONTINUE
19000   C
19100           DO 111 I=1,10
19200           Z=0.0
19300           DO 112 J=1,N1
19400           Z=Z+P(I,J)
19500   112     CONTINUE
19600           Z=Z-P(I,IHI)
19700           PBAR(I)=Z/RN
19800   111     CONTINUE
19900   C
20000   C-----------------------------------------------------------
20100   C
20200   C  REFLECTION THROUGH PBAR
20300   C
20400   1013    CONTINUE
20500   C
20600           WRITE(5,713)
20700   713     FORMAT(3X,'REFLECTION')
20800   C
20900           DO 16 I=1,10
21000   C
21100           PSTAR(I)=PBAR(I)+RCOEFF*( PBAR(I)-P(I,IHI) )
21200   C
21300   16      CONTINUE
21400           CALL SXFUNC(N,PSTAR,FVALUE)
21500           YSTAR=FVALUE
21600           ICOUNT=ICOUNT+1
21700   C
21800           WRITE(5,703)ICOUNT
21900           WRITE(5,704)(PSTAR(I),I=1,N)
22000           WRITE(5,705)YSTAR
22100   C
22200   703     FORMAT(3X,'ITERATION NO. = ',I5)
22300   704     FORMAT(3X,10(G12.5,X))
22400   705     FORMAT(3X,'FUNCTION VALUE = ',G15.6)
22500   C
22600   C-----------------------------------------------------------
22700   C
22800   C  CHECK REFLECTION
22900   C
23000           IF(YSTAR.GE.YLO)GO TO 1019
23100   C
23200   C-----------------------------------------------------------
23300   C
23400   C  SUCCESSFUL REFLECTION. TRY EXTENSION
23500   C
23600           WRITE(5,714)
23700   714     FORMAT(3X,'EXTENSION')
```

```
C
      DUM=ACOEFF+ECOEFF
      DO 18 I=1,N
C
      P2STAR(I)=PBAR(I)+DUM*(PBAR(I)-P(I,IHI))
C
18    CONTINUE
      CALL SIFUNC(N,P2STAR,FVALUE)
      Y2STAR=FVALUE
      ICOUNT=ICOUNT+1
C
      WRITE(5,703)ICOUNT
      WRITE(5,704)(P2STAR(I),I=1,N)
      WRITE(5,705)Y2STAR
C
C-----------------------------------------------------------
C
C  CHECK EXTENSION
C
      IF(Y2STAR.GE.YSTAR)GO TO 1018
C
C-----------------------------------------------------------
C
C  ACCEPT EXTENSION, Y2STAR IS NEW YLO
C
      DO 21 I=1,N
      P(I,IHI)=P2STAR(I)
21    CONTINUE
      Y(IHI)=Y2STAR
      GO TO 9999
C
C-----------------------------------------------------------
C
C  REJECT THE EXTENSION
C
1018  CONTINUE
C
      WRITE(5,717)
717   FORMAT(3X,'REJECT EXTENSION')
C     .
      DO 35 I=1,N
      P(I,IHI)=PSTAR(I)
35    CONTINUE
      Y(IHI)=YSTAR
      GO TO 9999
C
C-----------------------------------------------------------
C
C  NEW POINT NOT LOWEST
C
1019  CONTINUE
C
      IF(YSTAR.GE.YSECHI)GO TO 1021
C
C-----------------------------------------------------------
C
C  NEW POINT LESS THAN SECOND HIGH, SO ACCEPT
C
      DO 36 I=1,N
      P(I,IHI)=PSTAR(I)
36    CONTINUE
      Y(IHI)=YSTAR
      GO TO 9999
C
C-----------------------------------------------------------
C
C  NEW POINT .GE. SECOND HIGH, SO NOT ACCEPTABLE
C
1021  CONTINUE
C
      IF(YSTAR.GE.YHIGH)GO TO 1027
C
C-----------------------------------------------------------
C
C  CONTRACT TO POINT OUTSIDE SIMPLEX
C
      WRITE(5,719)
719   FORMAT(3X,'EXTERNAL CONTRACTION')
C
      DUM=CCOEFF*RCOEFF
      DO 22 I=1,10
```

```
C
      P2STAR(I)=PBAR(I)+DUM*(PBAR(I)-P(I,IHI))
C
22    CONTINUE
      CALL SIFUNC(N,P2STAR,FVALUE)
      Y2STAR=FVALUE
      ICOUNT=ICOUNT+1
C
      WRITE(5,703)ICOUNT
      WRITE(5,704)(P2STAR(I),I=1,N)
      WRITE(5,705)Y2STAR
C
C-----------------------------------------------------------
C
C  CHECK EXTERNAL CONTRACTION
C
      IF(Y2STAR.GE.YSTAR)GO TO 1026
C
C-----------------------------------------------------------
C
C  ACCEPT CONTRACTION
C
      WRITE(5,721)
721   FORMAT(3X,'ACCEPT CONTRACTION')
C
      DO 37 I=1,N
      P(I,IHI)=P2STAR(I)
37    CONTINUE
      Y(IHI)=Y2STAR
C
C-----------------------------------------------------------
C
      IF(Y2STAR.LT.YSECHI)GO TO 9999
C
C  CONTRACT SIMPLEX
      GO TO 1025
C
C-----------------------------------------------------------
C
C  REJECT EXTERNAL CONTACTION
C
1026  CONTINUE
C
      WRITE(5,722)
722   FORMAT(3X,'REJECT CONTRACTION')
C
      DO 38 I=1,N
      P(I,IHI)=PSTAR(I)
38    CONTINUE
C
C  CONTRACT SIMPLEX
      GO TO 1025
C
C-----------------------------------------------------------
C
C  TRY CONTRACTION TO INSIDE OF THE SIMPLEX
C
1027  CONTINUE
C
      WRITE(5,723)
723   FORMAT(3X,'INTERNAL CONTRACTION')
C
      DUM=CCOEFF*RCOEFF
      DO 24 I=1,10
C
      P2STAR(I)=PBAR(I)+DUM*(P(I,IHI)-PBAR(I))
C
24    CONTINUE
      CALL SIFUNC(N,P2STAR,FVALUE)
      Y2STAR=FVALUE
      ICOUNT=ICOUNT+1
C
      WRITE(5,703)ICOUNT
      WRITE(5,704)(P2STAR(I),I=1,N)
      WRITE(5,705)Y2STAR
C
C-----------------------------------------------------------
```

```
C
C  CHECK INTERNAL CONTRACTION
C
      IF(Y2STAR LT. YHIGH)GO TO 1029
C
C  CONTRACT SIMPLEX
      GO TO 1025
C
C-----------------------------------------------------------------
C
1029  CONTINUE
C
      WRITE(5,724)
724   FORMAT(3X,'ACCEPT INTERNAL CONTRACTION')
C
      DO 43 I=1,N
      P(I,IHI)=P2STAR(I)
43    CONTINUE
      Y(IHI)=Y2STAR
C
C-----------------------------------------------------------------
C
      IF(Y2STAR.LT.YSECHI)GO TO 9999
C
C  CONTRACT SIMPLEX
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
C  CONTRACTION OF THE SIMPLEX
C
1025  CONTINUE
C
      WRITE(5,725)
725   FORMAT(3X,'CONTRACTION OF SIMPLEX')
C
      WRITE(5,757)(P(I,ILO),I=1,N)
757   FORMAT(3X,'BEST PARS ',10(G12.5,X))
      WRITE(5,758)YLO
758   FORMAT(3X,'BEST FUNCTION VALUE=',G20.6)
C
      DO 27 JSIMP=1,N1
      DO 26 I=1,10
C
      P(I,JSIMP)=0.5*( P(I,JSIMP) + P(I,ILO) )
C
      PSTAR(I)=P(I,JSIMP)
26    CONTINUE
      CALL SIXFUNC(N,PSTAR,FVALUE)
      Y(JSIMP)=FVALUE
      ICOUNT=ICOUNT+1
27    CONTINUE
C
C-----------------------------------------------------------------
C
9999  CONTINUE
      GO TO 1002
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
C  SELECT THE BEST POINT FOUND BY THE ALGORITHM
C
900   CONTINUE
C
```

```
45200           YLO=Y(1)
45300           ILO=1
45400           DO 30 I=1,N1
45500           IF(Y(I) GE. YLO)GO TO 29
45100           YLO=Y(I)
45200           ILO=I
45300   29      CONTINUE
45400   30      CONTINUE
45500   C
45600           Y(ILO)=BIGNUM
45700   C
45800           YSEC=Y(1)
45900           ISEC=1
47000           DO 32 I=1,N1
47100           IF(Y(I) GE. YSEC)GO TO 31
47200           YSEC=Y(I)
47300           ISEC=I
47400   31      CONTINUE
47500   32      CONTINUE
47600   C
47700           Y(ILO)=YLO
47800   C
47900   C-----------------------------------------------------------------
48000   C
48100           IF(KIST. EQ. 1)WRITE(5,601)
48200           IF(KIST. EQ. 2)WRITE(5,602)
48300           IF(KIST. EQ. 3)WRITE(5,603)
48400   C
48500   601     FORMAT(2X,'*** FUNCTION CONVERGENCE ***')
48600   602     FORMAT(2X,'*** PARAMETER CONVERGENCE ***')
48700   603     FORMAT(3X,'*** ITERATION LIMIT EXCEEDED ***')
48800   C
48900   C-----------------------------------------------------------------
49000   C
49100           DO 34 I=1,N1
49200           A(I)=P(I,ILO)
49300   34      CONTINUE
49400   C
49500   C-----------------------------------------------------------------
49600   C
49700           WRITE(5,500)NPAR
49800           WRITE(5,510)ICOUNT
49900   C
50000   500     FORMAT(3X,'NUMBER OF PARAMETERS OPTIMIZED= ',I5)
50100   510     FORMAT(3X,'TOTAL NUMBER OF ITERATIONS USED= 'I5)
50200   C
50300           WRITE(5,520)YSEC
50400   520     FORMAT(3X,'NEXT TO BEST ESTIMATES .. FUNC VAL =',G20.6)
50500           WRITE(5,530)(P(I,ISEC),I=1,N)
50600   530     FORMAT(3X,10(G12.5,X))
50700   C
50800           WRITE(5,540)YLO
50900   540     FORMAT(3X,'BEST ESTIMATES .. FUNC VAL =',G20.6)
51000           WRITE(5,530)(P(I,ILO),I=1,N)
51100   C
51200   C-----------------------------------------------------------------
51300   C
51400           RETURN
51500           END
51600   C
51700   C*****************************************************************
```

```
C:::::::::::::ZZZZZZZZZZZZZZZZZZZZZZZZ::::::::::::::::::::::ZZZZZZ
C
      SUBROUTINE SIFUNC(NPAR,PARVAL.FUNC)
C
C-----------------------------------------------------------------
C
C  ***  COMPUTES THE FUNCTION VALUE FOR THE SIMPLEX ALGORITHM
C
C-----------------------------------------------------------------
C
      COMMON/CNTROL/ISERCH.IOP,ISENSE,IMODEL.IREPAR
      DIMENSION PARVAL(10)
C
C-----------------------------------------------------------------
C
C  ***  INVERT THE PARAMETERIZATION FROM PARVAL(I) TO VALUES OF
C  ***  THE MODEL PARAMETERS.
C
      CALL PARIZ2(PARVAL)
C
C-----------------------------------------------------------------
C
C  ***  COMPUTE FUNCTION VALUE
C
      IF(IMODEL.EQ.1)CALL MODLZ1
C
      IF(IMODEL.EQ.2)CALL MODLA1
C
      CALL LOGLIK(FUNC)
C
C-----------------------------------------------------------------
C
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE EIGER(NT,EIGVAL,EIGVEC)
C
C-----------------------------------------------------------------
C
      DOUBLE PRECISION A(10,10),D(10),E(10),AKO,EPS
C
      DIMENSION EIGVAL(10),EIGVEC(10,10)
C
C-----------------------------------------------------------------
C
      AKO=1.0D38
      EPS=0.000000001
C
C-----------------------------------------------------------------
C
      DO 100 I=1,NT
      DO 90  J=1,NT
      A(I,J) = EIGVEC(I,J)
90    CONTINUE
100   CONTINUE
C
C-----------------------------------------------------------------
C
      CALL SYMORD(A,D,E,AKO,NT,10,EPS,.FALSE.,.TRUE.,
     1 .FALSE.,IFAIL)
C
C-----------------------------------------------------------------
C
      CALL DSORT(NT,D,A)
C
C-----------------------------------------------------------------
C
      DO 200 I=1,NT
      EIGVAL(I) = D(I)
      DO 190 J=1,NT
      EIGVEC(I,J) = A(I,J)
190   CONTINUE
200   CONTINUE
C
C-----------------------------------------------------------------
C
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
C ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
      SUBROUTINE ANLYS2
C
C-------------------------------------------------------------
C
C ** CONDUCTS THE ANALYSIS OF THE PRECISION AND SENSITIVITY
C    OF THE PARAMETER ESTIMATES USING THE HESSIAN ESTIMATE
C
C-------------------------------------------------------------
C
C ** INFO REQD BY THE PROGRAM - NPAR,PARVAL(I),HESS(I,J)
C
C ** EXPLANATION OF VARIABLES
C
C         NPAR     - NO  OF PARAMETERS
C         PARVAL(I) - VALUES OF PARAMETERS
C         HESS(I,J) - ESTIMATE OF HESSIAN MATRIX AT PARVAL(I)
C         HESINV(I,J) - INVERSE OF HESSIAN
C         HDET     - DETERMINANT OF THE HESSIAN MATRIX
C         IHDUM1,IHDUM2 - DUMMY ARGUMENTS
C         CORREL(I,J) - CORRELATION MATRIX OF THE PAR ESTIMATES
C         GG(I,J),G(I) - MATRICES FOR CONFIDENCE LIMIT CALCULATION
C         GGINV(I,J)    -     "      "      "      "      "
C         ALPHA(I) - NPAR-DEGREE FREEDOM, APPROX. 95 % CONFIDENCE
C                         LIMITS FOR THE PARAMETERS
C         EIGVAL(I) - EIGENVALUES IN DIMINISHING SIZE ORDER
C         EIGVEC(I,J) - EIGENVECTORS
C         P2SIGM(I) - 1-DEG-FREEDOM, APPROX  95% CONFIDENCE LIMITS
C         PCOEFV(I) - 1-DEG-FREEDOM PARAMETER EST COEF OF VAR.
C         PNCOFV(I) - N-DEG-FREEDOM PARAMETER EST COEF OF VAR.
C         H2(I,J),EIGV2(I) - USED IN 2-PAR SENSITIVITY ANALYSIS
C         IHSIGN(I,J) - STORES INFO ON DEFINITENESS OF 2-PAR
C                             SUB-HESSIANS
C         ELONG(I,J) - STORES ELONG INDEX VALUES
C         THETA(I,J) - STORES THETA INDEX VALUES
C         RATL(I) - STORES AXIS LENGTH RATIOS
C         CORDET - DETERMINANT OF THE CORRELATION MATRIX
C         CONDNO - CONDITION NO. OF THE HESSIAN MATRIX
C
C============================================================
C
C============================================================
C
      COMMON/PARAMS/NPAR,UM,UK,BM,BK,Z,X,USI,BSI
      COMMON/ESTIM/PARVAL(10),SLOP(10),HESS(10,10)
      COMMON/A1/  HESINV(10,10),CORREL(10,10),HDET,CORDET,CONDNO
      COMMON/A2/ ALPHA(10),P2SIGM(10)
      COMMON/A3/ ELONG(10,10),THETA(10,10),RATL(10),EIGVEC(10,10)
      COMMON/A4/ IHSIGN(10,10),EIGVAL(10),PCOEFV(10),PNCOFV(10)
C
C
      DIMENSION IHDUM1(10),IHDUM2(10),EPSLON(10)
      DIMENSION H2(10,10),EIGV2(10)
      DIMENSION GG(9,9),G(9),GGINV(9,9)
C
      DATA (EPSLON(I),I=1,10)/3.84,5.99,7.81,9.49,11.07,12.59,
     1  14.07,15.51,16.92,18.31/
C
C============================================================
C
C
C ** DETERMINE THE INVERSE OF THE HESSIAN MATRIX
C
      DO 100 I=1,NPAR
      DO 110 J=1,NPAR
      HESINV(I,J)=HESS(I,J)
      EIGVEC(I,J)=HESS(I,J)
110   CONTINUE
100   CONTINUE
C
      NT=NPAR
      EPS=0.00000000001
      CALL ARRAY(2,NT,NT,10,10,HESINV,HESINV)
      CALL MINV(HESINV,NT,HDET,IHDUM1,IHDUM2,EPS,IFLAG)
      CALL ARRAY(1,NT,NT,10,10,HESINV,HESINV)
      IF(IFLAG.NE.1)GO TO 900
      WRITE(5,120)
120   FORMAT('  THE HESSIAN MATRIX IS SINGULAR ',/)
      ISING=1
C ** ( ISING=1 INDICATES HESSIAN IS SINGULAR )
      GO TO 998
```

```
07000    C
07100    400     CONTINUE
07200    C*****************************************************************
07300    C
07400    C ** COMPUTE THE CORRELATION MATRIX OF THE PARAMETER ESTIMATES
07500    C
07600            DO 130 I=1,NPAR
07700            DO 140 J=I,NPAR
07800            CORREL(I,J)=HESINV(I,J)/SQRT(HESINV(I,I)*HESINV(J,J))
07900            CORREL(J,I)=CORREL(I,J)
08000     140    CONTINUE
08100     130    CONTINUE
08200    C
08210            CORDET=1.0
08220            DO 54 I=1,NPAR
08230            CORDET=CORDET*HESINV(I,I)
08240    54      CONTINUE
08250            CORDET=1.0/(HDET*CORDET)
08260    C
08300    C*****************************************************************
08400    C
08500    C ** COMPUTE NPAR-DEGREE OF FREEDOM 95 % CONFIDENCE
08600    C    LIMITS FOR THE PARAMETERS BASED ON THEORY PROPOSED BY
08700    C    GUPTA (PHD. THESIS)
08800    C
08900            DO 150 JJ=1,NPAR
08905            J1=JJ-1
08910            J2=JJ+1
09000    C
09250            IF(JJ.EQ.1)GO TO 161
09300            DO 160 J=1,J1
09400            DO 170 K=J,J1
09500            GG(J,K)=HESS(J,K)
09600            GG(K,J)=GG(J,K)
09700     170    CONTINUE
09800     160    CONTINUE
09850     161    CONTINUE
09875            IF(JJ.EQ.NPAR) GO TO 221
09900            DO 180 J=J2,NPAR
10000            DO 190 K=J,NPAR
10100            GG(J-1,K-1)=HESS(J,K)
10200            GG(K-1,J-1)=HESS(J,K)
10300     190    CONTINUE
10400     180    CONTINUE
10450            IF(JJ.EQ.1)GO TO 221
10500            DO 200 J=1,J1
10600            DO 210 K=J2,NPAR
10700            GG(J,K-1)=HESS(J,K)
10800     210    CONTINUE
10900     200    CONTINUE
11000            DO 220 J=J2,NPAR
11100            DO 230 K=1,J1
11200            GG(J-1,K)=HESS(J,K)
11300     230    CONTINUE
11400     220    CONTINUE
11450     221    CONTINUE
11500    C
11600            DO 250 J=1,NPAR-1
11700            DO 260 K=J,NPAR-1
11800            GGINV(J,K)=GG(J,K)
11900            GGINV(K,J)=GG(J,K)
12000     260    CONTINUE
12100     250    CONTINUE
12200    C
12300            NT=NPAR-1
12400            EPS=0.00000000001
12500            CALL ARRAY(2,NT,NT,9,9,GGINV,GGINV)
12600            CALL MINV(GGINV,NT,GDET,IHDUM1,IHDUM2,EPS,IFLAG)
12700            CALL ARRAY(1,NT,NT,9,9,GGINV,GGINV)
12800            IF(IFLAG.NE.1)GO TO 910
12900            WRITE(5,270)JJ
13000    270     FORMAT(' THE  GG(',I2,') MATRIX IS SINGULAR ')
13100            ALPHA(JJ)=0.0
13200            GO TO 150
13300    C
13400     910    CONTINUE
13500            DO 280  J=1,J1
13600            G(J)=HESS(J,JJ)
13700     280    CONTINUE
13800            DO 290 J=J2,NPAR
12900            G(J-1)=HESS(J,JJ)
14000     290    CONTINUE
```

```
14100   C
14200           SUM=0.0
14300           DO 300 K=1,NPAR-1
14400           DO 310 J=1,NPAR-1
14500           SUM=SUM+G(K)*G(J)*GGINV(K,J)
14600    310    CONTINUE
14700    300    CONTINUE
14800   C
14900   C ** ALPHA(JJ) IS THE AMOUNT TO BE + OR - FROM THE PARAMETER
15000   C     ESTIMATE TO GET THE 95 % CONFIDENCE LIMITS
15100   C     PNCOFV(JJ) IS THE N-DEG-FREEDOM ANALOGY TO COEF OF VAR
15200   C
15300           ASUM=HESS(JJ,JJ)-SUM
15400           ALPHA(JJ) = SQRT( EPSLON(NPAR)/ASUM )
15500           PNCOFV(JJ)=0.5*ALPHA(JJ)/PARVAL(JJ)
15600   C
15700    150    CONTINUE
15800   C
15900   C*****************************************************************
16000   C
16100    888    CONTINUE
16200   C
16300   C*****************************************************************
16400   C
16500   C ** SOLVE FOR EIGENVALUES AND EIGENVECTORS
16600   C
16650           NT=NPAR
16700           CALL EIGER(NT,EIGVAL,EIGVEC)
16800   C
16820           CONDNC=EIGVAL(1)/EIGVAL(NPAR)
16860   C
16900   C*****************************************************************
17000   C
17100   C       CALCULATE SINGLE PARAMETER SENSITIVITIES
17200   C
17300           DO 370 I=1,NPAR
17400           PSIGM(I)=SQRT(3.84/HESS(I,I))
17500           PCOEFV(I)=0.5*PSIGM(I)/PARVAL(I)
17600    370    CONTINUE
17700   C
17800   C*****************************************************************
17900   C
18000   C       TWO PARAMETER SENSITIVITIES
18100   C
18200           NA=2
18300           I=1
18400    700    CONTINUE
18500           DO 400 J=NA,NPAR
18600   C
18700           H2(1,1)=HESS(I,I)
18800           H2(2,2)=HESS(J,J)
18900           H2(1,2)=HESS(I,J)
19000           H2(2,1)=HESS(J,I)
19100   C
19200           H11=H2(1,1)
19300           H22=H2(2,2)
19400   C               .
19500   C              .
19600   C
19650           NT=2
19700           CALL EIGER(NT,EIGV2,H2)
19800   C
19900   C       EIGV2(1)=MAX EIGVAL , EIGV2(2)=MIN EIGVAL
20000   C

20100           IHSIGN(I,J)=0
20200           IF(EIGV2(1) GT 0.0)IHSIGN(I,J)=IHSIGN(I,J)+1
20300           IF(EIGV2(1) LT 0.0)IHSIGN(I,J)=IHSIGN(I,J)-1
20400           IF(EIGV2(2) GT 0.0)IHSIGN(I,J)=IHSIGN(I,J)+1
20500           IF(EIGV2(2) LT 0.0)IHSIGN(I,J)=IHSIGN(I,J)-1
20550           IHSIGN(J,I)=IHSIGN(I,J)
20600   C
20700   C ** NOTE IHSIGN(I,J)=(-2,0,2) CORRESP TO (MAX,MIN,SADDLE PTS)
20850   C
20900           IF(IHSIGN(I,J) EQ.0)GO TO 666
21000   C
21100           ELONG(I,J)=SQRT(EIGV2(2)/EIGV2(1))
21150           ELONG(J,I)=ELONG(I,J)
21200   C
```

```
21300           JSIGN=0
21400           IF(H2(1,2) GT 0.0)JSIGN=JSIGN+1
21500           IF(H2(1,2) LT 0.0)JSIGN=JSIGN-1
21600           IF(H2(2,2) GT 0.0)JSIGN=JSIGN+1
21700           IF(H2(2,2) LT 0.0)JSIGN=JSIGN-1
21800   C
21900           IF(JSIGN GT 0)ANGLE=1.0
22000           IF(JSIGN EQ 0)ANGLE=-1.0
22100           IF(JSIGN LT 0)ANGLE=1.0
22200   C
22300           PARAT=SQRT( (EIGV2(2)-H11) / (EIGV2(2)-H22) )
22400           THET=ATAN( PARAT ) * 360.0 / 6.2831853
22500           THETA(I,J)=ANGLE * THET
22505           THETA(J,I)=THETA(I,J)
22600   C
22700   C
22800   C
22900   666     CONTINUE
23000   C
23100   400     CONTINUE
23200           I=I+1
23300           NA=NA+1
23400           IF(I.EQ.NPAR)GO TO 777
23500           GO TO 700
23600   777     CONTINUE
23700   C
23800   C
23900   C*****************************************************************
24000   C
24100   C       AXIS LENGTH RATIOS
24200   C
24300   C
24400           DO 780 I=1,NPAR
24500           IF(EIGVAL(I) LT 0.0)GO TO 750
24600           RATL(I)=SQRT(EIGVAL(I)/EIGVAL(1) )
24700           GO TO 780
24800    750    RATL(I)=-9.9
24900    780    CONTINUE
25000   C
25100   C       .
25200   C*****************************************************************
25300   C
25400   C ** PRINT OUT THE RESULTS OF THE SENSITIVITY ANALYSIS
25500   C
25600           CALL PRINT2(ISING)
25700   C
25800   C ** END OF SENSITIVITY ANALYSIS OF PARAMETERS
25900   C
26000           RETURN
26100           END
```

```
C
      SUBROUTINE PRINT2(ISING)
C
C*****************************************************************
C ** PRINTS OUT THE RESULTS OF THE SENSITIVITY ANALYSIS
C    OF THE PARAMETER ESTIMATES
C
C ** INFO REQD BY PROGRAM - ISING, HESINV(I,J), CORREL(I,J)
C                          PARVAL(I), PSIGM(I), PCOEFV(I)
C                          ALPHA(I), PNCOFV(I), ELONG(I,J)
C                          THETA(I,J), IHSIGN(I,J), RATL(I)
C                          EIGVAL(I), EIGVEC(I,J), HDET
C
C*****************************************************************
C
C
      COMMON/PARAMS/NPAR,UM,UK,BM,BK,Z,X,USI,BSI
      COMMON/ESTIM/PARVAL(10),SLOP(10),HESS(10,10)
      COMMON/A1/  HESINV(10,10),CORREL(10,10),HDET,CORDET,CONDNO
      COMMON/A2/ ALPHA(10),PSIGM(10)
      COMMON/A3/ ELONG(10,10),THETA(10,10),RATL(10),EIGVEC(10,10)
      COMMON/A4/ IHSIGN(10,10),EIGVAL(10),PCOEFV(10),PNCOFV(10)
C
C*****************************************************************
C
C*****************************************************************
C
C ** PRINT OUT THE RESULTS OF THE SENSITIVITY ANALYSIS
C
      WRITE(5,9)
 9    FORMAT(1X,/,50('*'))
      WRITE(5,10)
 10   FORMAT(20X,'RESULTS OF THE SENSITIVITY ANALYSIS')
      WRITE(5,9)
C
C **** PRINT THE HESSIAN MATRIX
C
      WRITE(5,889)
889   FORMAT('--------THE HESSIAN MATRIX--------',/)
      DO 888 I=1,NPAR
      WRITE(5,902)(HESS(I,J),J=1,I)
888   CONTINUE
C
C **** PRINT THE INVERSE HESSIAN MATRIX
C
      IF(ISING.EQ.1)GO TO 850
      WRITE(5,901)
 901  FORMAT('--------THE INVERSE HESSIAN MATRIX--------',/)
      DO 855 I=1,NPAR
      WRITE(5,902)(HESINV(I,J),J=1,I)
 902  FORMAT(5X,10(G11.5,1X),/)
 855  CONTINUE
C
C ** PRINT THE DETERMINANT OF THE HESSIAN MATRIX
      WRITE(5,12)HDET
 12   FORMAT(10X' DETERMINANT OF HESSIAN MATRIX = ',G20.10)
C
      WRITE(5,911)
 911  FORMAT(/,/,/)
C
C **** PRINT THE PARAMETER CORRELATION MATRIX
      WRITE(5,903)
 903  FORMAT(/,'--------THE PAR CORRELATION MATRIX--------',/)
      DO 856 I=1,NPAR
      WRITE(5,904)(CORREL(I,J),J=1,I)
 904  FORMAT(5X,10(G9.3,2X),/)
 856  CONTINUE
C
C ** PRINT THE DETERMINANT OF THE CORRELATION MATRIX
      WRITE(5,13)CORDET
13    FORMAT(10X,' DETERMINANT OF CORRELATION MATRIX = ',G20.10)
      WRITE(5,911)
C
C
```

```
C ** PRINT TABLE OF CONFIDENCE LIMITS
      WRITE(5,905)
  905 FORMAT(/,/,'-------- 95 % CONFIDENCE LIMITS --------')
  906 FORMAT(/,105('='))
      WRITE(5,907)
  907 FORMAT(32X,'1-DEGREE OF FREEDOM',22X,
     1 'N-DEGREES OF FREEDOM')
      WRITE(5,908)
  908 FORMAT(5X,'PARAMETER VALUE',10X,2('LOWER',15X,'UPPER',16X))
      WRITE(5,906)

      DO 857 I=1,NPAR
      Z1=PARVAL(I,+P2SIGM(I)
      Z2=PARVAL(I,+P2SIGM(I)
      Z3=PARVAL(I,-ALPHA(I)
      Z4=PARVAL(I,-ALPHA(I)
      WRITE(5,909)I,PARVAL(I),Z1,Z2,Z3,Z4
  909 FORMAT(5X,I2,3X,G10.5,4X,G15.9,5X,G15.9,6X,2(G15.9,5X))
  857 CONTINUE
C
      WRITE(5,906)
      WRITE(5,920)
  920 FORMAT(31X,2('COEFFICIENT OF VARIATION',18X))
      WRITE(5,906)
      DO 921 I=1,NPAR
      WRITE(5,922)I,PCOEFV(I),PNCOFV(I)
  922 FORMAT(5X,I2,27X,2(G15.9,26X))
  921 CONTINUE
      WRITE(5,906)
C
  850 CONTINUE
C
C ** PRINT 2-PAR SUBSPACE SENSITIVITY INDICES
C
      WRITE(5,911)
      WRITE(5,912)
  912 FORMAT('--------2-PAR SENSITIVITY INDICES--------',/,/)
      WRITE(5,913)
  913 FORMAT(10X,'ELONGATION MATRIX',/,10X,17('='),/)
      DO 914 I=1,NPAR
      WRITE(5,902)(ELONG(I,J),J=1,I)
  914 CONTINUE
C
      WRITE(5,911)
      WRITE(5,923)
  923 FORMAT(10X,'INTERACTION MATRIX (DEGREES)',/,10X,28('='),/)
      DO 924 I=1,NPAR
      WRITE(5,925)(THETA(I,J),J=1,I)
  925 FORMAT(5X,10(G8.3,4X),/)
  924 CONTINUE
C
      WRITE(5,911)
      WRITE(5,926)
  926 FORMAT(10X,'DEFINITNESS MATRIX  MAX=+2,MIN=+2,SADDLE'
     1 ' POINT=0',/,10X,47('='),/)
      DO 927 I=1,NPAR
      WRITE(5,928)(IHSIGN(I,J),J=1,I)
  928 FORMAT(5X,10(I5,5X),/)
  927 CONTINUE
C
C ** PRINT EIGVALS AND EIGVEC'S
C
      WRITE(5,911)
      WRITE(5,929)
  929 FORMAT('--------EIGENVALUES--------',/)
      WRITE(5,902)(EIGVAL(I),I=1,NPAR)
      WRITE(5,930)
  930 FORMAT('--------EIGENVECTORS--------',/)
      DO 931 I=1,NPAR
      WRITE(5,902)(EIGVEC(I,J),J=1,NPAR)
  931 CONTINUE
C
C ** PRINT CONDITION NUMBER OF HESSIAN
C
      WRITE(5,14)CONDNO
   14 FORMAT(10X,'CONDITION NUMBER = ',G20.10,/)
C
```

```
C** PRINT AXIS LENGTH RATIOS
C
        WRITE(5,911)
        WRITE(5,932)
 932    FORMAT(10X,'AXIS LENGTH RATIOS',/,10X,18('-'),/)
        WRITE(5,933)(I,RATL(I)),I=1,NFAR)
 933    FORMAT(10(10X,I2,5X,G11.5,/))
C
C ** END OF THE SENSITIVITY ANALYSIS PRINTOUT
C
        WRITE(5,906)
        WRITE(5,934)
 934    FORMAT(10X,'END OF SENSITIVITY ANALYSIS PRINTOUT')
        WRITE(5,906)
        WRITE(5,9)
C ***************************************************************
        RETURN
        END


C-------------------------------------------------------------------
C
        SUBROUTINE DSORT(N,A,B)
C
C-------------------------------------------------------------------
C
C  ***  ARRANGES VECTOR A(I) SO THAT ITS ELEMENTS ARE IN
C  ***  DECREASING ORDER OF MAGNITUDE.  AND THE COLUMNS OF
C  ***  B(I,J) THAT CORRESPOND TO THE ELEMENTS OF A(I)
C  ***  ARE ALSO ACCORDINGLY REARRANGED.
C
C-------------------------------------------------------------------
C
        DOUBLE PRECISION A(10),B(10,10),TEMP(10,10),T
        DIMENSION M(10)
C
C-------------------------------------------------------------------
C
        DO 10 I=1,N
        M(I)=I
10      CONTINUE
C
C-------------------------------------------------------------------
C
        K=N-1
        DO 16 I=1,K
        L=N-I
        DO 16 J=1,L
        IF( A(J) GT A(J+1) )GO TO 16
        T=A(J)
        A(J)=A(J+1)
        A(J+1)=T
        MT=M(J)
        M(J)=M(J+1)
        M(J+1)=MT
16      CONTINUE
C
C-------------------------------------------------------------------
C
        DO 200 I=1,N
        DO 100 J=1,N
        TEMP(I,J)=B(I,J)
100     CONTINUE
200     CONTINUE
C
        DO 400 J=1,N
        DO 300 I=1,N
        B(I,M(J)) = TEMP(I,J)
300     CONTINUE
400     CONTINUE
C
C-------------------------------------------------------------------
C
        RETURN
        END
```

```
C*****************************************************************
      SUBROUTINE ARRAY(MODE,I,J,N,M,S,D)                                ARAY
C  I BY J IS SIZE OF CURRENT MATRIX
C  N BY M IS DIMENSIONED SIZE OF MATRIX     M = N CURRENTLY
C  PURPOSE OF THIS SUBROUTINE IS TO CONVERT A 2-DIMENSIONAL MATRIX
C  D INTO A COMPACT 1-DIMENSIONAL ARRAY S (MODE = 2), AND BACK AGAIN
C  (MODE = 1)
      DIMENSION S(1),D(1)                                               ARAY
      NI = N - I                                                        ARAY
      IF(MODE-1) 100,100,120                                            ARAY
C                                                                       ARAY
C                                                                       ARAY
100   IJ = I * J + 1                                                    ARAY
      NM = N * J + 1                                                    ARAY
      DO 110   K=1,J                                                    ARAY
      NM=NM-NI                                                          ARAY
      DO 110  L=1,I                                                     ARAY
      IJ=IJ-1                                                           ARAY
      NM=NM-1                                                           ARAY
110   D(NM)=S(IJ)                                                       ARAY
      GO TO 140                                                         ARAY
C                                                                       ARAY
C                                                                       ARAY
120   IJ=0                                                              ARAY
      NM=0                                                              ARAY
      DO 130 K=1,J                                                      ARAY
      DO 125 L=1,I                                                      ARAY
      IJ=IJ+1                                                           ARAY
      NM=NM+1                                                           ARAY
125   S(IJ)=D(NM)                                                       ARAY
130   NM=NM+NI                                                          ARAY
140   RETURN                                                            ARAY
      END                                                               ARAY
CZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
C
C                                                                       DINV
C                                                                       DINV
C     ..................  ..................................................DINV
C                                                                       DINV
C        SUBROUTINE MINV                                                DINV
C                                                                       DINV
C        PURPOSE                                                        DINV
C           INVERT A MATRIX                                             DINV
C                                                                       DINV
C        USAGE                                                          DINV
C     CALL MINV(A,N,D,L,M,EPS,IFLAG)
C                                                                       DINV
C        DESCRIPTION OF PARAMETERS                                      DINV
C           A - INPUT MATRIX. DESTROYED IN COMPUTATION AND REPLACED BY  DINV
C               RESULTANT INVERSE.                                      DINV
C           N - ORDER OF MATRIX A                                       DINV
C           D - RESULTANT DETERMINANT                                   DINV
C           L - WORK VECTOR OF LENGTH N                                 DINV
C           M - WORK VECTOR OF LENGTH N                                 DINV
C  EPS - SMALLEST THAT PIVOT CAN BE WITHOUT BEING REGARDED AS SINGULAR
C   IFLAG = 0, NORMAL.   1, SINGULAR.
C                                                                       DINV
C        REMARKS                                                        DINV
C           MATRIX A MUST BE A GENERAL MATRIX                           DINV
C                                                                       DINV
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  DINV
C           NONE                                                        DINV
C                                                                       DINV
C        METHOD                                                         DINV
C           THE STANDARD GAUSS-JORDAN METHOD IS USED.  THE DETERMINANT  DINV
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT    DINV
C           THE MATRIX IS SINGULAR.                                     DINV
C                                                                       DINV
C     ..................................................................DINV
C                                                                       DINV
      SUBROUTINE MINV(A,N,D,L,M,EPS,IFLAG)
      DIMENSION A(1),L(1),M(1)                                          DINV
C                                                                       DINV
```

```
C         SEARCH FOR LARGEST ELEMENT                                    DINV
C                                                                       DINV
      IFLAG = 0
      DO 60 K=1,N                                                       DINV
      NK=(K-1)*N                                                        DINV
      L(K)=K                                                            DINV
      M(K)=K                                                            DINV
      KK=NK+K                                                           DINV
      BIGA=A(KK)                                                        DINV
      DO 20 J=K,N                                                       DINV
      IZ=N*(J-1)                                                        DINV
      DO 20 I=K,N                                                       DINV
      IJ=IZ+I                                                           DINV
   10 IF (ABS(BIGA)-ABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)                                                        DINV
      L(K)=I                                                            DINV
      M(K)=J                                                            DINV
   20 CONTINUE                                                          DINV
C                                                                       DINV
C         INTERCHANGE ROWS                                              DINV
C                                                                       DINV
      J=L(K)                                                            DINV
      IF(J-K) 35,35,25                                                  DINV
   25 KI=K-N                                                            DINV
      DO 30 I=1,N                                                       DINV
      KI=KI+N                                                           DINV
      HOLD=-A(KI)                                                       DINV
      JI=KI-K+J                                                         DINV
      A(KI)=A(JI)                                                       DINV
   30 A(JI) =HOLD                                                       DINV
C                                                                       DINV
C         INTERCHANGE COLUMNS                                           DINV
C                                                                       DINV
   35 I=M(K)                                                            DINV
      IF(I-K) 45,45,38                                                  DINV
   38 JP=N*(I-1)                                                        DINV
      DO 40 J=1,N                                                       DINV
      JK=NK+J                                                           DINV
      JI=JP+J                                                           DINV
      HOLD=-A(JK)                                                       DINV
      A(JK)=A(JI)                                                       DINV
   40 A(JI) =HOLD                                                       DINV
C                                                                       DINV
C         DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS       DINV
C         CONTAINED IN BIGA)                                            DINV
C                                                                       DINV
   45 IF (ABS(BIGA) GT EPS) GO TO 46
      IFLAG = 1
      RETURN                                                            DINV
   46 DO 55 I=1,N                                                       DINV
      IF(I-K) 50,55,50                                                  DINV
   50 IK=NK+I                                                           DINV
      A(IK)=A(IK)/(-BIGA)                                               DINV
   55 CONTINUE                                                          DINV
C                                                                       DINV
C         REDUCE MATRIX                                                 DINV
C                                                                       DINV
      DO 65 I=1,N                                                       DINV
      IK=NK+I                                                           DINV
      HOLD=A(IK)                                                        DINV
      IJ=I-N                                                            DINV
      DO 65 J=1,N                                                       DINV
      IJ=IJ+N                                                           DINV
      IF(I-K) 60,65,60                                                  DINV
   60 IF(J-K) 62,65,62                                                  DINV
   62 KJ=IJ-I+K                                                         DINV
      A(IJ)=HOLD*A(KJ)+A(IJ)                                            DINV
   65 CONTINUE                                                          DINV
C                                                                       DINV
```

```
      C         DIVIDE ROW BY PIVOT                                   DIN
      C                                                               DIN
            KJ=K-N                                                    DIN
            DO 75 J=1,N                                               DIN
            KJ=KJ+N                                                   DIN
            IF(J-K) 70,75,70                                          DIN
         70 A(KJ)=A(KJ)/SIGA                                          DIN
         75 CONTINUE                                                  DIN
      C                                                               DIN
      C         PRODUCT OF PIVOTS                                     DIN
      C                                                               DIN
            D = D*SIGA                                                
      C                                                               DIN
      C         REPLACE PIVOT BY RECIPROCAL                           DIN
      C                                                               DIN
            A(KK)=1.0/SIGA                                            DIN
         80 CONTINUE                                                  DIN
      C                                                               DIN
      C         FINAL ROW AND COLUMN INTERCHANGE                      DIN
      C                                                               DIN
            K=N                                                       DIN
        100 K=(K-1)                                                   DIN
            IF(K) 150,150,105                                         DIN
        105 I=L(K)                                                    DIN
            IF(I-K) 120,120,108                                       DIN
        108 JQ=N*(K-1)                                                DIN
            JR=N*(I-1)                                                DIN
            DO 110 J=1,N                                              DIN
            JK=JQ+J                                                   DIN
            HOLD=A(JK)                                                DIN
            JI=JR+J                                                   DIN
            A(JK)=-A(JI)                                              DIN
        110 A(JI) =HOLD                                               DIN
        120 J=K                                                       DIN
            IF(J-K) 100,100,125                                       DIN
        125 KI=K-N                                                    DIN
            DO 130 I=1,N                                              DIN
            KI=KI+N                                                   DIN
            HOLD=A(KI)                                                DIN
            JI=KI-K+J                                                 DIN
            A(KI)=-A(JI)                                              DIN
        130 A(JI) =HOLD                                               DIN
            GO TO 100                                                 DIN
        150 CONTINUE                                                  DIN
            RETURN                                                    DIN
            END                                                       DIN
```

```
            SUBROUTINE SYMORD(A, D, E, AKO, N, NA, EPS, AESCNV, VEC, TRD, FAIL)
      c
      c     from CACM algorithm 384, Eigenvalues and Eigenvectors of a
      c     Real Symmetric Matrix, by G.W. Stewart.
      c
      c     Explanation of the parameters in the calling sequence:
      c
      c     A - A two-dimensional array.  If the matrix is not initially
      c         tridiagonal, it is contained in the lower triangle of A.
      c         If eigenvectors are not requested the lower triangle of
      c         A is destroyed while the elements above the diagonal are
      c         left undisturbed.  If eigenvectors are requested, they
      c         are returned in the columns of A (double precision)
      c
      c     D - A one-dimensional array.  If the matrix is initially
      c         tridiagonal, D contains its diagonal elements.  On return
      c         D contains the eigenvalues of the matrix (double precision).
      c
      c     E - A one-dimensional array.  If the array is initially
      c         tridiagonal, E contains the off-diagonal elements.  Upon
      c         return E(i) contains the number of iterations required to
      c         compute the approximate eigenvalue D(i). (Double precision).
      c
      c     AKO - A double precision variable containing an original origin
      c         shift to be used until the computed shifts settle down.
      c         If AKO=1.0D38, the given or computed D(n) of the
      c         tridiagonal matrix will be used as the original origin shift.
      c
```

```
C     N - An integer variable containing the order of the matrix
C
C     NA - An integer variable containing the first dimension of the
C          matrix A.
C
C     EPS - A double precision variable containing a convergence
C          tolerance.
C
C     ABSCNV - A logical variable containing the value .true. if the
C          absolute convergence criterion is to be used or the value
C          .false. if the relative criterion is to be used.
C
C     VEC - A logical variable containing the value .true. if
C          eigenvectors are to be computed and returned in the array A
C          and otherwise containing the value .false..
C
C     TRD - A logical variable containing the value .true. if the
C          matrix is tridiagonal and located in the arrays D and E
C          and otherwise containing the value .false..
C
C     FAIL - An integer variable containing an error signal.  On
C          return the eigenvalues in D(fail+1)...D(n) and their
C          corresponding eigenvectors may be presumed accurate.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NA,N),D(N),E(N)
      DOUBLE PRECISION KQ,K1,K2,K,NINF,NORM,MAX
      INTEGER FAIL,SINCOS,RET
      LOGICAL ABSCNV,VEC,TRD,SHFT
      KQ=4KQ
      TITTER=50.0D0
      NM1=N-1
      NM2=N-2
      NINF=0.0D0
      ASSIGN 500 TO SINCOS
C
C     SIGNAL ERROR IF N IS NOT POSITIVE.
C
      IF (N.GT.0) GO TO 1
      FAIL=-1
      RETURN
C
C     SPECIAL TREATMENT FOR A MATRIX OF ORDER 1.
C
1     IF (N.GT.1) GO TO 5
      IF(.NOT.TRD) D(1)=A(1,1)
      IF(VEC) A(1,1)=1.0D0
      FAIL=0
      RETURN
C
C     IF MATRIX IS TRIDIAGONAL, SKIP THE REDUCTION.
C
5     IF(TRD) GO TO 100
      IF(N.EQ.2) GO TO 80
C
C     REDUCE THE MATRIX TO TRIDIAGONAL FORM BY HOUSEHOLDER'S METHOD.
C
      DO 70 L=1,NM2
      L1=L+1
      D(L)=A(L,L)
      MAX=0.0D0
      DO 10 I=L1,N
10    MAX=AMAX1(MAX,ABS(A(I,L)))
      IF(MAX.NE.0.0D0) GO TO 13
      E(L)=0.0D0
      A(L,L)=1.0D0
      GO TO 70
13    SUM=0.0D0
      DO 17 I=L1,N
      A(I,L)=A(I,L)/MAX
17    SUM=SUM+A(I,L)**2
```

```
61600           S2=SQRT(SUM)
61700           IF(A(L1,L) LT 0.0D0) S2=-S2
61800           E(L)=-S2*MAI
61900           A(L1,L)=A(L1,L)+S2
62000           A(L,L)=S2*A(L1,L)
62100           SUM1=0.0D0
62200           DO 50 I=L1,N
62300           SUM=0.0D0
62400           DO 20 J=L1,I
62500    20     SUM=SUM + A(I,J)*A(J,L)
62600           IF (I EQ.N) GO TO 40
62700           I1=I+1
62800           DO 30 J=I1,N
62900    30     SUM=SUM+A(J,L)*A(J,I)
63000    40     E(I)=SUM/A(L,L)
63100    50     SUM1=SUM1+A(I,L)*E(I)
63200           CON=0.5*SUM1/A(L,L)
63300           DO 60 I=L1,N
63400           E(I)=E(I)-CON*A(I,L)
63500           DO 60 J=L1,I
63600    60     A(I,J)=A(I,J)-A(I,L)*E(J) - A(J,L)*E(I)
63700    70     CONTINUE
63800    80     D(NM1)=A(NM1,NM1)
63900           D(N)=A(N,N)
64000           E(NM1)=A(N,NM1)
64100    C
64200    C    IF EIGENVECTORS ARE REQUIRED, INITIALIZE A.
64300    C
64400    100    IF(.NOT.VEC) GO TO 180
64500    C
64600    C    IF THE MATRIX WAS TRIDIAGONAL, SET A EQUAL TO THE IDENTITY MATRIX.
64700    C
64800           IF(.NOT.TRD.AND.N.NE.2) GO TO 130
64900           DO 120 I=1,N
65000           DO 110 J=1,N
65100    110    A(I,J)=0.0D0
65200    120    A(I,I)=1.0D0
65300           GO TO 180
65400    C
65500    C    IF THE MATRIX WAS NOT TRIDIAGONAL, MULTIPLY OUT THE
65600    C    TRANSFORMATIONS OBTAINED IN THE HOUSEHOLDER REDUCTION.
65700    C
65800    130    A(N,N)=1.0D0
65900           A(NM1,NM1)=1.0D0
66000           A(NM1,N)=0.0D0
66100           A(N,NM1)=0.0D0
66200           DO 170 L=1,NM2
66300           LL=NM2-L+1
66400           LL1=LL+1
66500           DO 140 I=LL1,N
66600           SUM=0.0D0
66700           DO 135 J=LL1,N
66800    135    SUM=SUM+A(J,LL)*A(J,I)
66900    140    A(LL,I)=SUM/A(LL,LL)
67000           DO 150 I=LL1,N
67100           DO 150 J=LL1,N
67200    150    A(I,J)=A(I,J)-A(I,LL)*A(LL,J)
67300           DO 160 I=LL1,N
67400           A(I,LL)=0.0D0
67500    160    A(LL,I)=0.0D0
67600    170    A(LL,LL)=1.0D0
67700    C
67800    C    IF AN ABSOLUTE CONVERGENCE CRITERION IS REQUESTED
67900    C    (ABSCNV=.TRUE.), COMPUTE THE INFINITY NORM OF THE MATRIX.
68000    C
68100    180    IF( NOT.ABSCNV) GO TO 200
68200           NINF=AMAX1(ABS(D(1))+ABS(E(1)),ABS(D(N))+ABS(E(NM1)))
68300           IF(N.EQ.2) GO TO 200
68400           DO 190 I=2,NM1
68500    190    NINF=AMAX1(NINF,ABS(D(I))+ABS(E(I))+ABS(E(I-1)))
68600    C
```

```
C   START THE QR ITERATION
C
200   NU=N
      NUM1=N-1
      SHFT=.FALSE.
      IF(K0.GT.1.0D37) K0=D(N)
      K1=K0
      TEST=NINF*EPS
      E(N)=0.0D0
C
C   CHECK FOR CONVERGENCE AND LOCATE THE SUBMATRIX IN WHICH THE
C   QR STEP IS TO BE PERFORMED.
C
210   DO 220 NNL=1,NUM1
      NL=NUM1-NNL+1
      IF(.NOT.ABSCNV) TEST=EPS*AMIN1(ABS(D(NL)),ABS(D(NL+1)))
      IF(ABS(E(NL)).LE.TEST) GO TO 230
220   CONTINUE
      GO TO 240
230   E(NL)=0.0D0
      NL=NL+1
      IF(NL.NE.NU) GO TO 240
      IF(NUM1.EQ.1) RETURN
      NU=NUM1
      NUM1=NU-1
      GO TO 210
240   E(NU)=E(NU)+1.0D0
      IF(E(NU).LE.TITTER) GO TO 250
      FAIL=NU
      RETURN
C
C   CALCULATE THE SHIFT.
C
250   CB=(D(NUM1)-D(NU))/2.0
      MAX=AMAX1(ABS(CB),ABS(E(NUM1)))
      CB=CB/MAX
      CC=(E(NUM1)/MAX)**2
      CD=SQRT(CB**2+CC)
      IF(CB.NE.0.0D0)CD=SIGN(CD,CB)
      K2=D(NU)-MAX*CC/(CB+CD)
      IF(SHFT) GO TO 270
      IF(ABS(K2-K1).LT.0.5*ABS(K2)) GO TO 260
      K1=K2
      K=K0
      GO TO 300
260   SHFT=.TRUE.
270   K=K2
C
C   PERFORM ONE QR STEP WITH SHIFT K ON ROWS AND COLUMNS NL
C   THROUGH NU
C
300   P=D(NL)-K
      G=E(NL)
      ASSIGN 310 TO RET
      GO TO SINCOS,(500)
310   DO 380 I=NL,NUM1
C
C   IF REQUIRED, ROTATE THE EIGENVECTORS
C
      IF (.NOT.VEC) GO TO 330
      DO 320 J=1,N
      TEMP=C*A(J,I)+S*A(J,I+1)
      A(J,I+1)=-S*A(J,I)+C*A(J,I+1)
320   A(J,I)=TEMP
```

```
C
C   PERFORM THE SIMILARITY TRANSFORMATION AND CALCULATE THE
C   NEXT ROTATION.
C
330   D(I)=C*D(I)-S*E(I)
      TEMP=C*E(I)-S*D(I+1)
      D(I+1)=-S*E(I)+C*D(I+1)
      E(I)=-S*K
      D(I)=C*D(I)+S*TEMP
      IF(I.EQ.NUM1) GO TO 380
      IF (ABS(S).GT.ABS(C)) GO TO 350
      R=S/C
      D(I+1)=-S*E(I)+C*D(I+1)
      P*D(I+1)-K
      Q=C*E(I+1)
      ASSIGN 340 TO RET
      GO TO SINCOS,(500)
340   E(I)=R*NORM
      E(I+1)=Q
      GO TO 380
350   P=C*E(I)-S*D(I+1)
      Q=S*E(I+1)
      D(I+1)=C*P/S+K
      E(I+1)=C*E(I+1)
      ASSIGN 360 TO RET
      GO TO SINCOS,(500)
360   E(I)=NORM
380   CONTINUE
      TEMP=C*E(NUM1)+S*D(NU)
      D(NU)=-S*E(NUM1)+C*D(NU)
      E(NUM1)=TEMP
      GO TO 210
C
C   INTERNAL PROCEDURE TO CALCULATE THE ROTATION CORRESPONDING TO
C   THE VECTOR (P,Q).
C
500   PP=ABS(P)
      QQ=ABS(Q)
      IF(QQ.GT.PP) GO TO 510
      NORM=PP*SQRT(1.0D0+(QQ/PP)**2)
      GO TO 520
510   IF(QQ.EQ.0.0D0) GO TO 530
      NORM=QQ*SQRT(1.0D0+(PP/QQ)**2)
520   C=P/NORM
      S=Q/NORM
      GO TO RET,(310,340,360)
530   C=1.0D0
      S=0.0D0
      NORM=0.0D0
      GO TO RET,(310,340,360)
      END
```

```
00100   C  Main prog to run GMAIN5/search in transf par space
00200   C
00250           IMPLICIT DOUBLE PRECISION(A-H,O-Z)
00300           DIMENSION PARO(10),PWR(10)
00400   C
00500           NPAR=2
00600           ITMAX=25
00700   C
00800           PARO(1)=20.0
00900           PARO(2)=6.0
00905   C
00910           PWR(1)=1.0
00920           PWR(2)=1.0
00930   C
00940           FNEPS=0.000001
00950           PAREPS=0.000001
00960           SDEPS1=0.001
00970           SDEPS2=0.01
00980   C
00990           ISER=1
01000   C
01100           CALL GMAIN5(NPAR,PARO,ITMAX,PAREPS,FNEPS,PWR,ISER,
01150          1                SDEPS1,SDEPS2)
01200   C
01300           STOP
01400           END

00100   C
00200   C
00300   C
00400   C   Oct 13 .  Corrected version of GMAIN4 with added steepest
00500   C             descent in both initial search and search for
00600   C             transformation parameter values
00700   C-------------------------------------------------------------
00800   C
00900   C
01000   C  Gradient Search using Transformations of the parameter
01100   C  space to improve quadratic convergence.
01200   C
01300           SUBROUTINE GMAIN5(NPAR,PARO,ITMAX,PAREPS,FNEPS,PWR,ISER,
01400          1                  SDEPS1,SDEPS2)
01500   C
01600   C-------------------------------------------------------------
01700           IMPLICIT DOUBLE PRECISION(A-H,O-Z)
01800   C
01900           DIMENSION SLOP(10),H(10,10),PARO(10),PAR1(10)
02000           DIMENSION TSLOP(10),TH(10,10),TPAR(10)
02100           DIMENSION PWR(10),RSLOP(10),RH(10,10),REPARO(10),REPAR1(10)
02200           DIMENSION RDEL(10)
02300   C
02400   C-------------------------------------------------------------
02500           WRITE(5,111)
02600    111    FORMAT(5X,'********** SUB GMAIN2 ******')
02700   C
02800   C  Initialise values
02900   C
03000           FLM=0.0
03100           IT=0
03200           ICH1=0
03300           ICH2=0
03400           ICH3=0
03500   C
03600           IF(ISER.EQ.1)GO TO 25
03700   C-------------------------- ----------------------------------
03800   C  Conduct an initial steepest descent search
03900   C
04000           INFO=1
04100           CALL MFUNC(NPAR,PARO,FZERO,SLOP,H,INFO)
04200           IT=IT+1
04300           WRITE(5,10)IT,(PARO(I),I=1,NPAR)
04400           WRITE(5,11)(SLOP(I),I=1,NPAR)
04500           WRITE(5,12)FZERO
04600    10     FORMAT(3X,'Iteration ',I5,/,3X,'Pars ',10(G12.6,X))
04700    11     FORMAT(3X,'Gradients ',10(G12.6,X))
04800    12     FORMAT(3X,'Fzero ',G15.6)
04900   C
05000           CALL STEEP(NPAR,PARO,PAR1,FZERO,FNEW,SLOP,IT,ITMAX,
05100          1                  SDEPS1,SDEPS2)
05200   C
```

```
05300           DO 30 I=1,NPAR
05400           PARO(I)=PAR1(I)
05500    30     CONTINUE
05600           FZERO=FNEW
05700           WRITE(5,13)
05800    13     FORMAT(3X,'*** Steepest Descent Over ***')
05900   C
06000   C-----------------------------------------------------------------
06100   C
06200   C  Begin Gradient Search
06300   C
06400   C
06500           INFO=1
06600           CALL MFUNC(NPAR,PARO,FZERO,SLOP,H,INFO)
06700           IT=IT+1
06800   C
06900   C
07000    555    CONTINUE
07100   C   Main loop of the program begins here
07200   C
07300   C      Store best par values,gradients and hessian
07400           DO 100 I=1,NPAR
07500           TPAR(I)=PARO(I)
07600           TSLOP(I)=SLOP(I)
07700           DO 110 J=1,NPAR
07800           TH(I,J)=H(I,J)
07900    110    CONTINUE
08000    100    CONTINUE
08100   C
08200   C  Print out values of interest
08300   C
08400           WRITE(5,600)IT,(FWR(I),I=1,NPAR)
08500    600    FORMAT(/,60('*'),/,9X,'ITER ',I3,/,5X,'TRANSFORMATION PARS
08600          1   ',/,5X,10(G13.6,X))
08700           WRITE(5,810)(TPAR(I),I=1,NPAR)
08800    810    FORMAT(5X,'PAR ',10(G12.6,X))
08900           WRITE(5,815)(TSLOP(I),I=1,NPAR)
09000    815    FORMAT(5X,'SLOP ',10(G12.6,X))
09100           WRITE(5,816)FZERO
09200    816   FORMAT(9X,'FZERO=  ',G15.6)
09300   C
09400   C   Compute transformed par values, gradients, and hessian
09500   C
09600           CALL GTRAN1(NPAR,PWR,PARO,REPARO)
09700           CALL GTRAN2(NPAR,PWR,SLOP,RSLOP,H,RH,PARO)
09800   C
09900   C
10000   C  Compute appropriate quadratic step in the parameter(transform-
10100   C  med) space
10200   C
10300           CALL QNSTEP(NPAR,RH,RSLOP,RDEL,PLM)
10400   C
10500   C  Initially set step equal to 1.0
10600           STP=1.0
10700   C
10800    777    CONTINUE
10900   C  Control returns here when a step is rejected and a smaller
11000   C   step size is to be tried
11100   C
11200   C    compute new values in the transformed parameter space
11300   C
11400           DO 120 I=1,NPAR
11500           DD=STP*RDEL(I)
11600           REPAR1(I)=REPARO(I)-DD
11700    120    CONTINUE
11800   C
11900   C  Compute parameter value in original par space
12000   C
12100           CALL GTRAN3(NPAR,PWR,REPAR1,PAR1)
12200   C
12300           DO 550 I=1,NPAR
12400           IF(PAR1(I).LT.0.0)PAR1(I)=0.0
12500    550    CONTINUE
12600   C  Compute function, gradient and hessian at new iteration
12700   C
12800           INFO=1
12900           CALL MFUNC(NPAR,PAR1,FNEW,SLOP,H,INFO)
13000           IT=IT+1
13100           WRITE(5,817)FNEW
13200    817    FORMAT(5X,'FNEW=  ',G15.6)
13300   C
```

```
C  check exceedance of iteration limit
C
      IF(IT GE ITMAX)ICH=1
C
C  Check for parameter value change convergence
      I=1
  499 CONTINUE
      A=DABS( PAR1(I)-PARO(I) )/PARO(I)
      IF(A GT PAREPS)GO TO 501
      IF(I GE NPAR)GO TO 502
      I=I+1
      GO TO 499
C
  502 CONTINUE
C  Parameters have converged
      ICH2=1
C
C
  501 CONTINUE
C
C  Check if the new function value is acceptable
C
C
C  Decide on step acceptance or rejection
C
      IF(FNEW LT FZERO)GO TO 300
      WRITE(5,701)
  701 FORMAT(5X,'STEP REJECTED')
      IF(ICH1 EQ 1)GO TO 999
      IF(ICH2 EQ 1)GO TO 998
      STP=STP/2.0
      GO TO 777
C
  300 CONTINUE
      WRITE(5,702)
  702 FORMAT(5X,'STEP ACCEPTED')
C  Compute new values of the transformation parameters
C
      CALL GTRAN4(NPAR,FZERO,FNEW,PARO,PAR1,SLOP,H,PWR)
C
     *  DO 310 I=1,NPAR
      PARO(I)=PAR1(I)
  310 CONTINUE
C
C  Test for function convergence
      A=(FZERO-FNEW)/FZERO
      IF(A LE FNEPS)ICH3=1
C
  511 CONTINUE
C
      FZERO=FNEW
      IF(ICH1 EQ 1)GO TO 999
      IF(ICH2 EQ 1)GO TO 998
      IF(ICH3 EQ 1)GO TO 997
C
      GO TO 555
C
C-----------------------------------------------------------------
C
C  Termination of the search
C
  997 CONTINUE
      WRITE(5,802)
  802 FORMAT(5X,'*** Termination due to function convergence')
      GO TO 990
  998 CONTINUE
      WRITE(5,803)
  803 FORMAT(5X,'*** Termination due to parameter convergence')
      GO TO 990
  999 CONTINUE
      WRITE(5,804)
  804 FORMAT(5X,'*** Termination due to iter limit exceedance')
  990 CONTINUE
C
C  Print final values of the parameters, gradient vector and
C  the information matrix.
C
```

```
      WRITE(5,851)(TPAR(I),I=1,NPAR)
  851 FORMAT(5X,'Final parameters',/,5X,5(G15.6,5X),/,10X,
     1 5(G15.6,5X),/)
      WRITE(5,852)(TSLOP(I),I=1,NPAR)
  852 FORMAT(5X,'Gradients',/,5X,5(G15.6,5X),/,10X,5(G15.6,
     1 5X),/)
      DO 853 I=1,NPAR
      WRITE(5,854)(TH(I,J),J=1,I)
  854 FORMAT(5X,'Information matrix',5X,10(G11.6,X) )
  853 CONTINUE
C
C
C  If required perform the parameter sensitivity analysis
C
C     IF(ISENSE.EQ.1)CALL CANLYS(NPAR,TPAR,TH)
C
C-----------------------------------------------------------------
C  Last lines of program follow
      RETURN
      END
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
      SUBROUTINE GTRAN1(NPAR,PWR,P,RP)
C
C  Computes transformed parameter values
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION PWR(10),P(10),RP(10)
C
      DO 100 I=1,NPAR
      IF(DABS(PWR(I)).LE.0.001)GO TO 50
      RP(I)=(P(I)**PWR(I)-1.0)/PWR(I)
      GO TO 60
   50 CONTINUE
      RP(I)=DLOG(P(I))
      PWR(I)=0.0
   60 CONTINUE
  100 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
      SUBROUTINE GTRAN2(NPAR,PWR,SLOP,RSLOP,H,RH,P)
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C  Computes transformed gradients and hessian
C
      DIMENSION PWR(10),SLOP(10),RSLOP(10),H(10,10),RH(10,10)
      DIMENSION P(10)
C
      DO 100 I=1,NPAR
      RSLOP(I)=SLOP(I)*P(I)**(1.0-PWR(I))
      DUM=H(I,I)*P(I)**(2.0-2.0*PWR(I))
      RH(I,I)=DUM+SLOP(I)*(1.0-PWR(I))*P(I)**(1.0-2.0*PWR(I))
      DO 110 J=1,NPAR
      RH(I,J)=H(I,J)*(P(I)**(1.0-PWR(I)))*(P(J)**(1.0-PWR(J)))
      RH(J,I)=RH(I,J)
  110 CONTINUE
  100 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------
```

```
C-----------------------------------------------------------------
C
C
      SUBROUTINE CTRAN3(NPAR,PWR,RP,P)
C
C  Retransforms pars back to original space
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION PWR(10),RP(10),P(10)
C
      DO 100 I=1,NPAR
      IF(DABS(PWR(I)).LE.0.001)GO TO 50
      P(I)=( RP(I)*PWR(I)+1.0 )**(1.0/PWR(I))
      GO TO 60
 50   CONTINUE
      P(I)=DEXP( RP(I) )
      PWR(I)=0.0
 60   CONTINUE
 100  CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------

C-----------------------------------------------------------------
C
C
      SUBROUTINE CTRAN4(NPAR,F0,F1,P0,P1,SLOP,H,PWR)
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION P0(10),P1(10),SLOP(10),H(10,10),PWR(10)
      DIMENSION RS(10),RH(10,10),RP1(10),RP0(10),PWR1(10)
      DOUBLE PRECISION DEH(10,10),DERR(10),ERR,DELPWR(10)
C
C  Set initial step size
      STEP=1.0
      ITPWR=0
      ISE=0
C
 100  CONTINUE
C  Compute error in quadratic fit
C      First compute transformed values of pars,gradients,hess
      CALL CTRAN1(NPAR,PWR,P0,RP0)
      CALL CTRAN1(NPAR,PWR,P1,RP1)
      CALL CTRAN2(NPAR,PWR,SLOP,RS,H,RH,P1)
C      Compute squared prediction error with given transf.
      CALL ERROR(NPAR,RP0,RP1,RS,RH,F0,F1,SGERRA,ERRA)
      ITPWR=ITPWR+1
      WRITE(5,10)SGERRA
 10   FORMAT(5X,'SGERRA ',X,G12.6)
C
 555  CONTINUE
C
      IF(ISE EQ.1)GO TO 600
C-----------------------------------------------------------------
      WRITE(5,20)ITPWR
 20   FORMAT(3X,'ITPWR=',I1,3X,'Steepest descent for PWR(I)')
C
C Compute normalised gradient vector of quad fit wrt. pwr pars
      CALL GPWR(NPAR,P0,P1,RP0,RP1,PWR,RS,RH,ERRA,DELPWR)
      WRITE(5,11)(DELPWR(I),I=1,NPAR)
 11   FORMAT(5X,'DELPWR(I) ',10(G12.6,X))
      GO TO 700
C
C-----------------------------------------------------------------
C
 600  CONTINUE
C
      STEP=1.0
      ISA=0
      WRITE(5,21)ITPWR
 21   FORMAT(3X,'ITPWR=',I3,3X,'Gradient search for PWR(I)')
C
      CALL GPWR1(NPAR,P0,P1,RP0,RP1,PWR,RS,RH,ERRA,DERR,DEH)
C
C Conduct a  gradient search
C
      CALL GNSTEP(NPAR,DEH,DERR,DELPWR,0.0)
      WRITE(5,17)(DELPWR(I),I=1,NPAR)
 17   FORMAT(5X,'DELPWR(I) ',10(G12.6,X))
```

```
C
C-----------------------------------------------------------------
 700    CONTINUE
C
C
 250    CONTINUE
        DO 220 I=1,NPAR
        DD=STEP*DELPWR(I)
        PWR1(I)=PWR(I)+DD
        IF(PWR1(I).GT.15.0)PWR1(I)=15.0
        IF(PWR1(I).LT.-10.0)PWR1(I)=-10.0
 220    CONTINUE
        WRITE(5,12)(PWR1(I),I=1,NPAR)
 12     FORMAT(5X,'PWR1(I) ',/,10(G12.6,X))

C  Compute error with new transformation pars
        CALL GTRAN1(NPAR,PWR1,P0,RP0)
        CALL GTRAN1(NPAR,PWR1,P1,RP1)
        CALL GTRAN2(NPAR,PWR1,SLOP,RS,H,RH,P1)
        CALL ERROR(NPAR,RP0,RP1,RS,RH,F0,F1,SGERRB,ERRB)
        ITPWR=ITPWR+1
        WRITE(5,13)SGERRB
 13     FORMAT(5X,'SGERRB ',G12.6)
        IF(ITPWR.GT.20)GO TO 750
C
C  Decide to reject or accept step
C
        IF(SGERRB.LT.SGERRA)GO TO 300
C
        IF(ISEF.EQ.0)GO TO 800
        IF(ISA.EQ.0)GO TO 800
        GO TO 100
 800    CONTINUE
C
C  Reduce step size
        STEP=STEP/2.0
        IF(STEP.LT.0.01)GO TO 400
        GO TO 250
C
 400    CONTINUE
C
        IF(ISER.EQ.1)GO TO 750
C       End of steepest descent
C**************
        GO TO 750
C**************
C
        ISER=1
        GO TO 100
C
 750    CONTINUE
C Terminate search for new transformation pars
        WRITE(5,15)(PWR(I),I=1,NPAR)
 15     FORMAT(5X,'BEST PWR VALS',/,5X,10(G12.6,X))
        RETURN
C
 300    CONTINUE
C Step accepted.try new step
        SG=(SGERRA-SGERRB)/SGERRA
        IF(SG.LE.0.001)GO TO 750
        IF(SGERRB.LE.0.00001)GO TO 750
        ISA=1
        DO 310 I=1,NPAR
        PWR(I)=PWR1(I)
 310    CONTINUE
        SGERRA=SGERRB
        STEP=STEP*2.0
        GO TO 250
        END
C
C*****************************************************************
C*****************************************************************
C
```

```
41700   C
41800           SUBROUTINE ERROR(NPAR,A0,A1,RS,RH,F0,F1,SGERR,ERR)
41900   C
41950           IMPLICIT DOUBLE PRECISION(A-H,O-Z)
42000           DIMENSION A0(10),A1(10),RS(10),RH(10,10)
42100           DOUBLE PRECISION ERR
42200   C
42300   C
42400           Y=0.0
42500           Z=0.0
42600           DO 100 I=1,NPAR
42700           C=A0(I)-A1(I)
42800           Y=Y+RS(I)*C
42900    100    CONTINUE
43000           DO 110 I=1,NPAR
43100           DO 120 J=1,NPAR
43200           Z=Z+(A0(I)-A1(I))*RH(I,J)*(A0(J)-A1(J))
43300    120    CONTINUE
43400    110    CONTINUE
43500           ERR=F1-F0+Y+0.5*Z
43600           SGERR=ERR*ERR
43700           RETURN
43800           END
43900   C
44000   C*****************************************************************
44100   C*****************************************************************
44200   C
44300   C
44400           SUBROUTINE CPWR(NPAR,P0,P1,RP0,RP1,PWR,RS,RH,ERR,DERR)
44500   C
44600   C
44650           IMPLICIT DOUBLE PRECISION(A-H,O-Z)
44700           DIMENSION RS(10),RH(10,10)
44800           DOUBLE PRECISION SUM,SUM1,SUM2,DRS,AA,DC,ERR,DERR(10)
44900           DOUBLE PRECISION DRH(10,10),C(10)
45000           DIMENSION P0(10),P1(10),RP0(10),RP1(10),PWR(10)
45100   C
45200           DO 100 I=1,NPAR
45300           C(I)=RP0(I)-RP1(I)
45400    100    CONTINUE
45500   C Compute gradients wrt power parts
45600           DO 500 I=1,NPAR
45700           SUM=0.0
45800           DRS=-DLOG(P1(I))*RS(I)
45900           AA=DLOG(P0(I))*P0(I)**PWR(I)-DLOG(P1(I))*P1(I)**PWR(I)
46000           DC=( AA-C(I) )/PWR(I)
46100           SUM=SUM+DRS*C(I)+RS(I)*DC
46200   C
46300           SUM1=0.0
46400           DO 200 K=1,NPAR
46500           SUM1=SUM1+RH(I,K)*C(K)
46600    200    CONTINUE
46700           SUM=SUM+DC*SUM1
46800   C
46900           SUM2=0.0
47000           DO 210 K=1,NPAR
47100           IF(K.EQ.I)GO TO 220
47200           DRH(I,K)=-DLOG(P1(I))*RH(I,K)
47300           GO TO 230
47400    220    CONTINUE
47500           DRH(I,K)=-2.0*DLOG(P1(I))*RH(I,I)-(RS(I)/(P1(I)**PWR(I)))
47600    230    CONTINUE
47700           SUM2=SUM2+C(K)*DRH(I,K)
47800    210    CONTINUE
47900           SUM=SUM+C(I)*SUM2
48000           SUM=SUM-0.5*DRH(I,I)*C(I)*C(I)
48010           IF(ERR.GT.0.0)E=1.0
48020           IF(ERR.LT.0.0)E=-1.0
48050           DERR(I)=SUM*E
48100   C
48200   C
48300    500    CONTINUE
48400   C
```

```
C Normalise the gradient vector
      DO 505 I=1,NPAR
      IF(DABS(DERR(I)) GT 0.000000000001)GO TO 506
  505 CONTINUE
      DO 507 I=1,NPAR
      DERR(I)=0.0
  507 CONTINUE
      GO TO 508
  506 CONTINUE
C
  618 CONTINUE
      DO 610 I=1,NPAR
      IF(DABS(DERR(I)) GT.1000000.0)GO TO 612
  610 CONTINUE
      GO TO 614
  612 CONTINUE
      DO 616 I=1,NPAR
      DERR(I)=DERR(I)/1000000.0
  616 CONTINUE
      GO TO 618
C
  614 CONTINUE
      SUM=0.0
      DO 600 I=1,NPAR
      SUM=SUM+DERR(I)*DERR(I)
  600 CONTINUE
      SUM=DSQRT(SUM)
C
      DO 617 I=1,NPAR
      DERR(I)=-DERR(I)/SUM
617   CONTINUE
  508 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
      SUBROUTINE GNSTEP(NPAR,H,SLOP,DELPAR,PLM)
C     ~~~~~~~~~~~~~~~~~
C
C  Note - no normalisation
C
C*************************************************************************
C  Computes a Levenberg-Marquardt step in the parameter space.
C  Part of a Gradient algorithm to minimise a function of ten
C  or less parameters using successive quadratic approximations.
C
C  The Levenberg-Marquardt par. PLM is specified through the
C  calling program. If PLM=0.0, a Newton step in the subspace
C  of dominant eigenvalues results. By setting PLM not equal to
C  zero (positive values only) we constrain the size of the
C  step in an attempt to restrict it to a region in which the
C  quadratic approximation is more valid.
C
C  Also specified through the calling program are the following:
C       NPAR    = no. of parameters
C       H(i,j)  = information matrix
C       SLOP(i) = matrix of first partial derivatives wrt. pars.
C
C  The following information is returned to the calling program
C       DELPAR(i) = recommended change in the par values
C                   to get a reduction in the function
C
C*************************************************************************
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION H(10,10),U(10,10)
      DIMENSION HEIG(10),Z(10,10),SLOP(10)
      DOUBLE PRECISION DELPAR(10)
C
      DATA EPS/0.0000000001/,EPS2/0.00000001/,AKO/1.0D38/
C
C-----------------------------------------------------------------------
```

```
C
C
C
      DO 110 I=1,NPAR
      DO 120 J=1,NPAR
      U(I,J)=H(I,J)
  120 CONTINUE
  110 CONTINUE
C
C  ***  Compute the Eigenvalues and Eigenvectors of H
C
C
      NT=NPAR
      CALL EIGSYD(U,HEIG,NT,NT,NT,10,0.5)
      DO 998 I=1,NPAR
      WRITE(5,997)HEIG(I)
  997 FORMAT(5X,'**** LAMDA  = ',3X,G20.6)
  998 CONTINUE
C
C  ***  Compute the (PSEUDO?) inverse of the Eigenvector matrix
C-----------------------------------------------------------------------
C  Check to ensure Hessian is positive definite
C
      DO 400 I=1,NPAR
      IF(HEIG(I).LT.0.0) HEIG(I)=DABS( HEIG(I) )
  400 CONTINUE
C  Find largest eigenvalue
      LMAX=HEIG(1)
      DO 410 I=2,NPAR
      IF(HEIG(I).GT.LMAX) LMAX=HEIG(I)
  410 CONTINUE
C
C-----------------------------------------------------------------------
C
C
C
C  ***  Count no  of eigvals larger than EPS*(max HEIG(I))
C
      NEIG=NPAR
      CUTOFF=EPS*( HEIG(1) )
      DO 150 I=2,NPAR
      IF(HEIG(I).GT.CUTOFF) GO TO 150
      NEIG=I-1
      GO TO 160
  150 CONTINUE
  160 CONTINUE
C
C  *** Compute a Levenberg-Marquardt(?) step with given PLM
C
C  *** Compute the trial step DELPAR
C
      WRITE(5,99)NPAR,NEIG
   99 FORMAT(5X,'NPAR=',I2,2X,'NEIG=',I2)
C
      DO 170 I=1,NEIG
      IF( HEIG(I).LE.CUTOFF )GO TO 165
      A=0.0
      DO 180 J=1,NPAR
      A=A+U(J,I)*SLOP(J)
  180 CONTINUE
      DO 190 J=1,NPAR
      B(J,I)=A*U(J,I)
  190 CONTINUE
  165 CONTINUE
  170 CONTINUE
C
C
      PLM1=PLM+HEIG(1)
      DO 200 I=1,NPAR
      DELPAR(I)=0.0
      DO 210 J=1,NEIG
      IF( HEIG(J).LE.CUTOFF )GO TO 175
      DELPAR(I)=DELPAR(I)-( B(I,J)/(HEIG(J)+PLM1) )
  175 CONTINUE
  210 CONTINUE
  200 CONTINUE
C
C
      WRITE(5,900)(DELPAR(I),I=1,NPAR)
  900 FORMAT(2X,'DELPAR',X,10(G12.6,X))
      DO 910 I=1,NPAR
```

```
c3900            WRITE(6,920)(RH(I,J),J=1,NPAR)
c4000     920    FORMAT(2X,'RHESS',3,10(G12.6,X))
c4100     910    CONTINUE
c4200            RETURN
c4300     C  *** Last line of subroutine follows
c4400            END
c4500     C
c4600     C*****************************************************************
c4700     C*****************************************************************
c4800     C
c4900     C
c5000            SUBROUTINE GPWR1(NPAR,PO,P1,RPO,RP1,PWR,RS,RH,ERR,DF,DFH)
c5100     C
c5200     C---------------------------------------------------------------
c5300     C
c5400     C  Subroutine to compute gradient and hessian of quadratic fit
c5500     C  with respect to the power transformation parameters
c5600     C
c5700     C---------------------------------------------------------------
c5750            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
c5800            DIMENSION PO(10),P1(10),RPO(10),RP1(10),PWR(10),RS(10)
c5900            DIMENSION RH(10,10)
c6000            DOUBLE PRECISION ALO(10),AL1(10),ARO(10),AR1(10),AI(10),
c6100           1  DAI(10),DAII(10),DC(10),D2C(10),BII(10),DBII(10),D2BII(10),
c6200           2  SUMO,SUM1,SUM2,DBIK,D2BIK,DUM1,DUM2,DUM3,DUM4,DUM5,DUM6,
c6300           3  ERR(10),DFH(10,10),B(10,10),C(10),DFH(10,10),DF(10)
c6400     C
c6500     C
c6600            DO 100 I=1,NPAR
c6700     C
c6800            ALO(I)=DLOG(PO(I))
c6900            AL1(I)=DLOG(P1(I))
c7000            ARO(I)=PO(I)**PWR(I)
c7100            AR1(I)=P1(I)**PWR(I)
c7200            AI(I)=RS(I)
c7300            DAI(I)=-AL1(I)*AI(I)
c7400            DAII(I)=-AL1(I)*DAI(I)
c7500            C(I)=RPO(I)-RP1(I)
c7600            DC(I)=(ALO(I)*ARO(I)-AL1(I)*AR1(I)-C(I))/PWR(I)
c7700            D2C(I)=(ALO(I)*ALO(I)*ARO(I)-AL1(I)*AL1(I)*AR1(I)-2.0*
c7800           1  DC(I))/PWR(I)
c7900            BII(I)=RH(I,I)
c8000            DBII(I)=-2.0*AL1(I)*BII(I)-(AI(I)/AR1(I))
c8100            D2BII(I)=-2.0*AL1(I)*(DBII(I)-(AI(I)/AR1(I)))
c8200            DO 105 J=1,NPAR
c8300            B(I,J)=RH(I,J)
c8400     105    CONTINUE
c8500     C
c8600     100    CONTINUE
c8700     C
c8800            DO 110 I=1,NPAR
c8900            SUMO=0.0
c9000            SUM1=0.0
c9100            SUM2=0.0
c9200     C
c9300     C
c9400
c9500            DO 120 K=1,NPAR
c9600            IF(K.EQ.I)GO TO 130
c9700            DBIK=-AL1(I)*B(I,K)
c9800            D2BIK=B(I,K)*AL1(I)**2
c9900            GO TO 140
70000     130    CONTINUE
70100            DBIK=DBII(K)
70200            D2BIK=D2BII(K)
70300     140    CONTINUE
```

```
           SUM0=SUM0+C(K)*B(I,K)
           SUM1=SUM1+C(K)*DBIK
           SUM2=SUM2+C(K)*D2BIK
  120      CONTINUE
C
           DUM1=DAI(I)+SUM1-0.5*C(I)*DBII(I)
           DUM2=AI(I)+SUM0
           DERR(I)=C(I)*DUM1+DC(I)*DUM2
C
           DUM3=DAII(I)+SUM2+0.5*DC(I)*DBII(I)-0.5*C(I)*D2BII(I)
           DUM4=2.0*(DAI(I)+SUM1)+DBII(I)*(DC(I)-0.5*C(I))
           DEH(I,I)=C(I)*DUM3+DC(I)*DUM4+D2C(I)*DUM2
C
C
           IF(I.EQ.NPAR)GO TO 160
           DO 150 K=I+1,NPAR
           DBIK1=-AL1(I)*B(I,K)
           DBIK2=-AL1(K)*B(I,K)
           D2BIK=AL1(I)*AL1(K)*B(I,K)
           DUM5=DC(K)*DBIK1+C(K)*D2BIK
           DUM6=DC(K)*B(I,K)+C(K)*DBIK2
           DEH(I,K)=C(I)*DUM5+DC(I)*DUM6
           DEH(K,I)=DEH(I,K)
  150      CONTINUE
  160      CONTINUE
  110      CONTINUE
C
           DO 200 I=1,NPAR
           DO 210 J=1,NPAR
           DFH(I,J)=2.0*( ERR*DEH(I,J)+DERR(I)*DERR(J)  )
  210      CONTINUE
           DF(I)=2.0*ERR*DERR(I)
  200      CONTINUE
C
C
           RETURN
           END
C**************************************************************
C**************************************************************
C
```

```
00100   C
00200   C  Program to conduct a steepest descent search
00300   C
00400   C-----------------------  --------------------------------
00500   C
00600           SUBROUTINE STEEP(NPAR, P0, P1, F0, F1, SLOP, IT, ITMAX, PEPS1,
00700          1                         PEPS2)
00800   C
00850           IMPLICIT DOUBLE PRECISION(A-H, O-Z)
00900           DIMENSION P0(10), P1(10), SLOP(10), PT0(10), DEL(10), H(10, 10)
01000   C
01100   C  Initialise
01200           ICH1=0
01300           ICH2=0
01400   C
01500           DO 100 I=1, NPAR
01600           PT0(I)=P0(I)
01700    100    CONTINUE
01800   C
01900           FT0=F0
02000   C
02100           WRITE(5, 700)
02200    700    FORMAT(3X, '*** Steepest descent search ***')
02300   C
02350           STEP=1.0
02400    444    CONTINUE
02500   C
02600   C  Compute unit direction vector
02700   C
02800           SUM=0.0
02900           DO 110 I=1, NPAR
03000           SUM=SUM-SLOP(I)*SLOP(I)
03100    110    CONTINUE
03200           SUM=DSQRT( SUM )
03300           DO 120 I=1, NPAR
03400           DEL(I)=-SLOP(I)/SUM
03500    120    CONTINUE
03600   C
03700           WRITE(5, 702)(DEL(I), I=1, NPAR)
03800    702    FORMAT(3X, 'Del ', 10(G12.6, X))
03900   C
04000   C
04100    555    CONTINUE
04200   C
04300   C  Conduct a steepest descent search
04400   C
04600    666    CONTINUE
04700           DO 130 I=1, NPAR
04800           DD=STEP*DEL(I)
04900           P1(I)=PT0(I)+DD
05000    130    CONTINUE
05100           INFO=0
05200           CALL MFUNC(NPAR, P1, F1, SLOP, H, INFO)
05300           IT=IT+1
05400           WRITE(5, 704)IT, STEP, (P1(I), I=1, NPAR)
05500           WRITE(5, 706)F1
05600    704    FORMAT(3X, 'Iter ', I5, 2X, 'STEP ', G15.6, /, 3X 'Pars ', 10(G12.6, X))
05700    706    FORMAT(3X, 'F1   ', G15.6)
05800   C
05900   C  Check for convergence
06000   C
06100           IF(IT.GE.ITMAX)ICH1=1
06200           I=1
06300    499    CONTINUE
```

```
C
C  Termination of steepest descent search
C
 999    CONTINUE
        WRITE(5,804)
 804    FORMAT(3X,'*** Term. of steep  desc. - Iter Limit exceeded')
        GO TO 990
C
 998    CONTINUE
        WRITE(5,802)
 802    FORMAT(3X,'*** Term  of steep  desc. - Par Convergence')
C
 990    CONTINUE
        WRITE(5,808)(PTO(I),I=1,NPAR)
        WRITE(5,809)(SLOP(I),I=1,NPAR)
        WRITE(5,810)FTO
 808    FORMAT(3X,'Final pars ',/,3X,10(G12.6,X))
 809    FORMAT(3X,'Gradients ',/,3X,10(G12.6,X))
 810    FORMAT(3X,'Function = ',G15.8)
C
        DO 170 I=1,NPAR
        P(I)=PTO(I)
 170    CONTINUE
        F=FTO
        RETURN
        END
C
C
```

APPENDIX 3

Conditional Parameter Sensitivity; Concentricity and Interaction

In this appendix, four common measures of parameter sensitivity are presented.

A-3.1 Continued Parameter Sensitivity [$CPS_i(\theta)$]

This index is the one used most commonly in parameter sensitivity analysis. The conditional parameter sensitivity of parameter O_i at θ ($CPS_i(\theta)$ in Figure A-3.1) is the amount that the parameter can vary while remaining inside the indifference region while the other parameters are held fixed. The index does not account for the compensating effects of the simultaneous variation of two or more parameters on the model output. Let s_{ij} be the $(_{ij})$th element of the matrix $\nabla_\theta^2 SI(\theta)$. Then, setting $\{\nabla\theta\}_j = 0$ in (10) $j \neq$, solution for the maximum variation of $\{\nabla\theta\}$ yields:

$$CPS_i(\theta) = \pm \left[S_{ii}^{\Psi}\right]^{1/2} \qquad \text{(A-3.1)}$$

A-3.2 Concentricity (X_{ij}) and Interaction (I_{ij})

The conditional parameter sensitivity index discussed above measures only individual parameter sensitivity. Sorooshian and Arfi (1982) introduced two indices that measure two-parameter concurrent interdependence. These indices deal with the shapes of the elliptical regions in two-parameter subspaces formed by holding all but the two parameters of interest constant [i.e., $\{\nabla\theta\}_k = 0$ $k \neq (i, j)$]. The first index, called "concentricity" (X), characterizes the deviation of the shape of the ellipse from a circle by the ratio of the lengths of the major and minor axes of the ellipse.

Hence,

$$X_{ij} = \left[\frac{\lambda_{min}^{ij}}{\lambda_{max}^{ij}} \right]^{1/2} \qquad \text{(A-3.2)}$$

where λ_{min}^{ij} and λ_{max}^{ij} are the smaller and larger eigenvalues of the two-parameter submatrix of $\nabla_\theta^2 SI(\theta)$ given by:

$$S(i,j) = \begin{bmatrix} S_{ii} & S_{ij} \\ S_{ji} & S_{jj} \end{bmatrix} \qquad \text{(A-3.3)}$$

X varies between 0 and 1, where 1 indicates a perfect circle.

The second index, called interaction (I), measures the minimum of the two angles made by the major axis of the submatrix and the parameter axis directions. It is given by:

$$I_{ij} = \frac{v_{ij}}{45} \qquad \text{(A-3.4)}$$

where v_{ij} is the minimum of the absolute values of the two angles in degrees. Hence, I varies between 0 and 1 with 1 indicating the maximum angle of interaction.

If the value of X is small ($X << 1.0$) and I is large ($I \simeq 1.0$), this indicates a poor ability to distinguish the effects of perturbations of the two parameters on the output. For details, see Sorooshian and Arfi (1982).

A-3.3 Condition Number and Multiparameter Interaction

Two other useful indices of the shape and orientation of the ellipse are the condition number (CN) and the index of multi-parameter interaction (MPI). The condition number [see, for example, Bard (1974)] of a matrix is given by the ratio of the largest to smallest of its eigenvalues.

$$CN = \frac{\text{largest eigenvalue}}{\text{smallest eigenvalue}} \tag{A-3.5}$$

If an eigenvalue is zero, then CN = ∞ and the matrix is said to be ill-conditioned. (It will be positive semidefinite, and hence the model is structurally nonidentifiable). Geometrically, the eigenvalues are related in inverse proportion to squares of the lengths of the hyperellipse axis. A zero eigenvalue is associated with an axis of infinite length, with equal SI value at each point along the axis. The most desirable value is CN = 1 and the least desirable is CN = ∞.

A measure of MPI was introduced by Nathanson and Saidel (1982). It consists of evaluating the determinant of the normalized inverse of the $\nabla_\theta^2 SI(\theta)$ matrix:

$$MPI = \det\left(\text{norm } \nabla_\theta^2 SI(\theta)^{-1}\right) \tag{A-3.6}$$

where

$$\{\text{norm } \nabla_\theta^2 SI(\theta)^{-1}\}_{ij} = \{\nabla_\theta^2 SI(\theta)^{-1}\}_{ij} / [\{\nabla_\theta^2 SI(\theta)^{-1}\}_{ii} \{\nabla_\theta^2 SI(\theta)^{-1}\}_{jj}]^{1/2} \quad \text{(A-3.7)}$$

The diagonal elements of the normalized inverse of the $\nabla_\theta^2 SI(\theta)$ matrix all have value 1, and the off-diagonal elements lie between -1 and +1. (If $\nabla_\theta^2 SI(\theta)^{-1}$ were a covariance matrix, the normalized matrix would be the correlation matrix). In the absence of parameter interaction, the off-diagonal terms of the normalized matrix will be zero and MPI = 1. As interaction increases, the value of MPI reduces until the model is non-identifiable, i.e., MPI = 0. Hence, it is desirable that the value of MPI be close to 1.0.

APPENDIX 4

DERIVATION OF PARAMETER SENSITIVITY INDEX

APPENDIX 4

The equation of the bounding hyperellipse is given by (3.10). Let $\Delta\theta = x$, $\nabla_\theta^2 SI(\theta) = H$, and $\Psi = \varepsilon$. Substituting in (3.10), we have:

$$x^T \cdot H \cdot x = \varepsilon \tag{A-4.1}$$

We wish to solve for the extreme points (χ_i, max) of this hyperellipse in the various parameter (χ_i) directions.

The extreme points result from solutions of:

$$\frac{\partial \varepsilon}{\partial x} = 0 \rightarrow Hx = 0 \tag{A-4.2}$$

Hence,

$$\begin{aligned} \frac{\partial \varepsilon}{\partial \chi_1} &= 0 = h_{11}\chi_1 + h_{12}\chi_2 + \cdots + h_{1p}\chi_p \\ &\vdots \\ \frac{\partial \varepsilon}{\partial \chi_p} &= 0 = h_{p1}\chi_1 + h_{p2}\chi_2 + \cdots + h_{pp}\chi_p \end{aligned} \tag{A-4.3}$$

where h_{ij} is the ij^{th} element of the matrix H, and χ_i is the i^{th} element of the vector x.

For χ_j to be a maximum, the p - 1 equations $\partial\varepsilon/\partial\chi_i = 0$ ($i \neq j$) must be solved simultaneously for p unknowns. This is equivalent to:

$$
\begin{array}{l}
j^{th}\ \text{row removed} \\
\left[\begin{array}{cccc}
h_{11} & h_{12} & \cdots & h_{1p} \\
\vdots & & & \\
h_{i1} & h_{i2} & \cdots & h_{ip} \\
\hline
h_{k1} & h_{k2} & \cdots & h_{kp} \\
\vdots & & & \\
h_{p1} & h_{p2} & \cdots & h_{pp}
\end{array}\right]
\end{array}
\quad
\left[\begin{array}{c} x_1 \\ \\ \vdots \\ \\ x_p \end{array}\right]
= \{0\} = R_j x
\qquad (p-1) \times 1
\qquad (A\text{-}4.4)
$$

$(p - 1) \times p$ $\qquad$ $p \times 1$

The solution of this equation will represent the vector passing through the points where x_j is a maximum for a given ε. We can rewrite Equation (A-4.4) as : (no j^{th} row or column)

$$
\left[\begin{array}{ccccccc}
h_{11} & \cdots & h_{1i} & h_{1k} & \cdots & h_{1p} \\
\vdots & & \vdots & \vdots & & \\
h_{i1} & \cdots & h_{ii} & h_{ik} & \cdots & h_{ip} \\
\hline
h_{k1} & \cdots & h_{ki} & h_{kk} & \cdots & h_{kp} \\
 & & \vdots & \vdots & & \\
h_{p1} & \cdots & h_{pi} & h_{pk} & \cdots & h_{pp}
\end{array}\right]
\left[\begin{array}{c} x_1 \\ \vdots \\ x_i \\ \hline x_k \\ \vdots \\ x_p \end{array}\right]
= -x_j
\left[\begin{array}{c} h_{1j} \\ \vdots \\ h_{ij} \\ \hline h_{kj} \\ \vdots \\ h_{pj} \end{array}\right]
\qquad (A\text{-}4.5)
$$

$(p - 1) \times (p - 1)$ $\qquad$ $(p - 1) \times 1$ $\qquad$ $(p - 1) \times 1$

Since there are $(p - 1)$ simultaneous equations and p unknowns to be solved for in Equation (A-4.5), we can arbitrarily specify one of the unknowns.

The most convenient choice here is $x_j = 1$. Substituting into (A-4.5) and rewriting in matrix vector notation:

$$G_J \cdot \tilde{x}_J = -g_J \qquad \text{(A-4.6)}$$

where

GJ = matrix obtained from H by removing the j^{th} row and column;

gJ = vector obtained from the j^{th} column of H by removing the j^{th} element; and

$\tilde{x}$ = vector obtained from x by removing the j^{th} element.

Therefore,

$$\tilde{x}_J = -G_J^{-1} \cdot g_J \qquad \text{(A-4.7)}$$

if G_J^{-1} exists. The vector we are seeking is x_J^* where:

$$x_J^{*T} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_i, \tilde{x}_j = 1, \tilde{x}_k, \ldots, \tilde{x}_p\} \qquad \text{(A-4.8)}$$

The scaled value of x_J^* is the unit vector in the direction passing through $x_{j,max}$ and is the quantity of interest. Define M_J as the matrix obtained from a p-dimensional identity matrix by removing the j^{th} row and n_J as the j^{th} column of a p-dimensional identity matrix.

Then:

$$R_J = M_J H \tag{A-4.9}$$

$$G_J = R_J M_J^T = M_J H M_J \tag{A-4.10}$$

$$g_J = R_J \cdot n_J = M_J \cdot H \cdot n_J \tag{A-4.11}$$

Therefore,

$$\chi_J = -G_J^{-1} \cdot g_J = -(M_J \cdot H \cdot M_J^T)^{-1}(M_J \cdot H \cdot n_J) \tag{A-4.12}$$

Now, x_J^* is the direction vector intercepting the ellipse at the required point. Let α_j be the value of χ_j at which the direction vector X_J^* intercepts the ellipse; i.e., χ_j represents the maximum extent that α_j can vary positively along its axis while remaining within the ellipse. Hence,

$$\chi_j,max = \alpha_j$$

and the vector x_J^* intercepts the ellipse at the point:

$$x_J^{**} = \alpha_j \cdot x_J^* \tag{A-4.13}$$

at which point the ellipse has the value:

$$\epsilon = x_J^{**T} \cdot H \cdot x_J^{**} \qquad \text{A-4.15}$$

$$= (\alpha_j)^2 x_J^{*T} \cdot H \cdot x_J^* \tag{A-4.16}$$

$$= (\alpha_j)^2 [\tilde{x}_1, \ldots, \tilde{x}_i, \tilde{x}_k, \ldots, \tilde{x}_p, 1] \left[\begin{array}{c|c} G_J & g_j \\ \hline g_J^T & h_{jj} \end{array}\right] \left[\begin{array}{c} \tilde{x}_1 \\ \vdots \\ \tilde{x}_i \\ \hline \tilde{x}_k \\ \vdots \\ \tilde{x}_p \\ 1 \end{array}\right] \tag{A-4.17}$$

$$= (\alpha_j)^2 [\tilde{x}_J^T \mid 1] \left[\begin{array}{c|c} G_J & g_J^T \\ \hline g_J & j_{jj} \end{array}\right] \left[\begin{array}{c} -\tilde{x}_J \\ \hline 1 \end{array}\right] \tag{A-4.18}$$

$$\varepsilon = (\alpha_j)^2 \left[\tilde{x}_J^T G_J \tilde{x}_J + 2\tilde{x}_J^T g_J + h_{jj}\right. \tag{A-4.19}$$

Substituting for $\tilde{x}_J$ from Equation (A-4.7):

$$\varepsilon = (x_j)^2 [g_J^T G_J^{-1} G_J G_J^{-1} g_J - 2 g_J^T G_J^{-1} g_J + h_{jj}] \tag{A-4.20}$$

$$= (\alpha_j)^2 [h_{jj} - g_J^T G_J^{-1} g_J] \tag{A-4.21}$$

Hence,

$$\alpha_j = \chi_{j,max} = \pm [\epsilon/(h_{jj} - g_J{}^T G_J{}^{-1} g_J)]^{1/2} \qquad \text{(A-4.22)}$$

Substituting $\epsilon = \Psi$ and $H = \nabla_\theta{}^2 SI(\theta)$,

$$\chi_{j,max} = \pm [\Psi/(s_{jj} - g_J{}^T G_J{}^{-1} g_J)]^{1/2} \qquad \text{(A-4.23)}$$

Hence, the parameter sensitivity index $PS_i(\theta)$ is:

$$PS_i(\theta) = \pm[\Psi/(s_{ii} - g_i{}^T G_i{}^{-1} g_i)]^{1/2} \qquad \text{(A-4.24)}$$

APPENDIX 5

MODIFICATION OF HORTON'S INFILTRATION EQUATION FOR USE WITH A CONCEPTUAL RAINFALL-RUNOFF MODEL

Horton's infiltration equation is:

$$f_t = f_{min} + [f_{max} - f_{min}]e^{-Kt} \qquad \text{(A-5.1)}$$

where

f_t = infiltration demand at time t;

f_{min} = minimum infiltration demand ($t = \infty$);

f_{max} = maximum infiltration demand ($t = 0$); and

K = rate parameter ($K > 0$).

This equation gives infiltration demand as a function of time. It can be represented diagramatically as shown in Figure A-5.1.

Horton's equation gives instantaneous infiltration demand at a fixed time t, measured from the time at which the infiltration process began (e.g., the beginning of the rainfall event). If it is assumed that the receiving zone (into which the water is infiltrating) is empty at time = 0, and that sufficient water is always available to meet demand, then Equation (A-5.1) represents actual rate of infiltration at time t. When $t = \infty$, the rate of infiltration is minimum (= f_{min}), and corresponds to the receiving zone being full. The value of f_{min} is not equal to zero, indicating that water drains continually from the receiving zone, thus making room for water to enter even when the receiving zone is full.

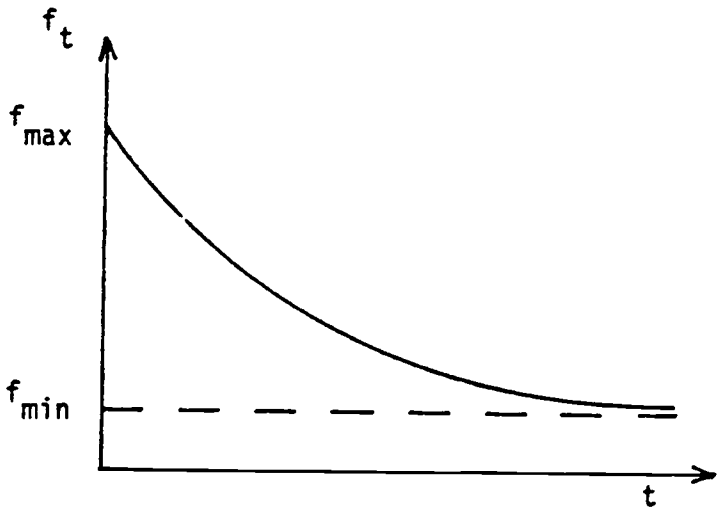


Figure A-5.1. Diagramatic representation of Horton's equation

A conceptual model is generally programmed on a digital computer and, usually, time is not involved explicitly as a variable in the computations of moisture transfers. Instead, it is more convenient to relate the moisture transfer computations to the current states of moisture storage in various parts of the model. Hence, Equation (A-5.1) is in a form unsuitable for use in most conceptual models. A formulation which is more suitable may be derived in the following manner. First, the cumulative infiltration volume during a fixed size computational interval Δt will be derived as a function of time t.

Let Δt = fixed size computational time interval;

F_n = cumulative infiltration volume during the n^{th} time interval from $(n - \frac{\Delta t}{2})$ to $(n + \frac{\Delta t}{2})$.

From Figure A-5.2, we can see that:

$$F_n = \text{area under the infiltration curve from } (n - \tfrac{\Delta t}{2}) \text{ to } (n + \tfrac{\Delta t}{2})$$

$$\simeq f_n \cdot \Delta t \qquad \text{(A-5.2)}$$

$$\simeq f_{min} \cdot \Delta t + (f_{max} - f_{min})\, \Delta t\, e^{-Kn} \qquad \text{(A-5.3)}$$

$$\simeq F_{min} + (F_{max} - F_{min})e^{-Kn} \qquad \text{(A-5.4)}$$

where

$F_{min} = f_{min} \cdot \Delta t$ = constant

$F_{max} = f_{max} \cdot \Delta t$ = constant

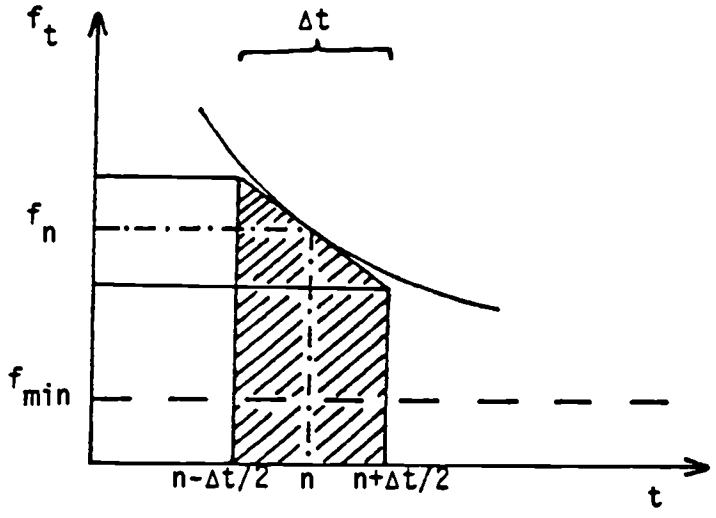


Figure A-5.2. Cumulative infiltration during time interval $(n - \frac{\Delta t}{2})$ to $(n + \frac{\Delta t}{2})$

Equation (A-5.4) represents volumes of infiltration during different fixed-size intervals of time. The equation must be transformed to represent infiltration volume in terms of the volume of water present in the lower zone.

Let UZM = maximum capacity of receiving zone; and

UZC_n = contents of receiving zone at beginning of computational time interval 'n'.

Note that when n = 0, UZC_n = 0, and that when n = ∞, UZC_n = UZM. Hence, a one-to-one transformation of the variable n (range 0 - ∞) to the variable UZC_n (range 0 - UZM) is required. One suitable transformation is:

$$n = \frac{UZC}{UZM-UZC} \qquad \text{(A-5.5)}$$

Hence, Equation A-5.4 can be transformed to:

$$F_n = F_{min} + [F_{max} - F_{min}] \cdot \exp\{-K\,(\frac{UZC_n}{UZM-UZC_n})\} \qquad \text{(A-5.6)}$$

The above equation is represented diagramatically in Figure A-5.3.

Now, the shape of the curve given by Equation (A-5.6) is limited to being of exponential form. A more general formulation allowing more flexibility to the shape of the curve can be obtained by replacing:

$$\exp\ -K(\frac{UZC_n}{UZM-UZC_n}) \qquad \text{(A-5.7)}$$

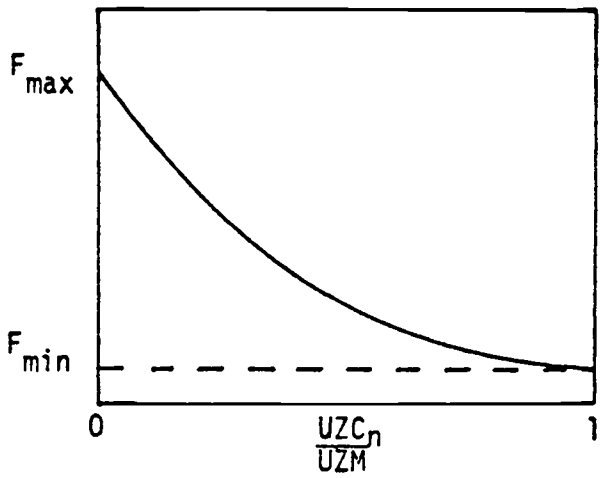


Figure A-5.3. Diagramatic representation of Equation (A-5.6)

in Equation (A-5.6) by:

$$\left(\frac{UZM-UZC_n}{UZM}\right)^X \qquad \text{(A-5.8)}$$

where X = parameter controlling the shape of curve (X > 1).

Equation (A-5.8) has a general behavior similar to that of Equation (A-5.7) for values of X > 1. Substituting (A-5.8) for (A-5.7) in Equation (A-5.6), we obtain:

$$F_n \simeq F_{min} + (F_{max} - F_{min}) \cdot \left(\frac{UZM - UZC_n}{UZM}\right)^X \qquad \text{(A-5.9)}$$

The above equation is represented diagramatically in Figure A-5.4.

As discussed in Chapter 3, the Sacramento Model percolation equation has a form similar to Equation (A-5.9).

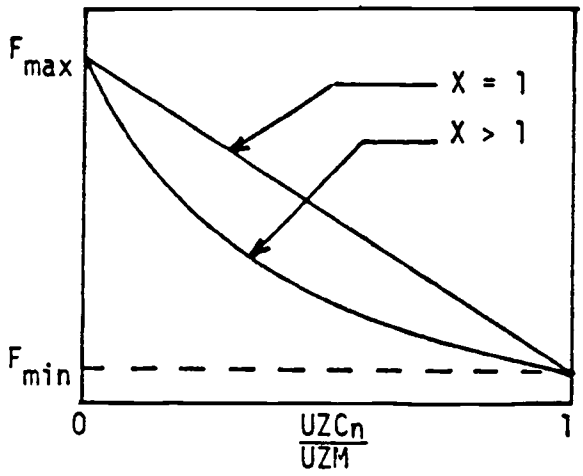


Figure A-5.4. Diagramatic representation of Equation (A-5.9)

APPENDIX 6

MEASURES OF CLOSENESS BASED ON LIKELIHOOD

A-6.1 The Simple Least Squares Function (SLS)

The SLS function is the minimum variance unbiased estimator when the errors have zero mean, are uncorrelated, and have a stationary variance. The estimator is:

$$\underset{\text{wrt } \theta}{\text{Min}} \quad SLS = \sum_{t=1}^{n} [z_t - z_t(\underline{\theta})]^2 \qquad \text{(A-6.1)}$$

where

$\tilde{z}_t$ = measured output at time t;

$z_t(\underline{\theta})$ = model output at time t for parameters; and

n = length of data set.

The SLS is the maximum likelihood estimator for the case of Gaussian, zero mean, uncorrelated, stationary variance noise.

A-6.2 The Maximum Likelihood Estimator for the Case of Gaussian First-Lag-Autocorrelated Errors (AMLE)

The AMLE (from Sorooshian, Gupta, and Fulton, 1981) is the minimum variance unbiased estimator when the errors have zero mean and stationary variance but are autocorrelated to first lag. Such errors may often arise in streamflow (or other) data when measurements are taken at relatively short intervals (e.g., hourly). The estimator is:

$$\underset{\text{wrt } \theta}{\text{Min}} \quad AMLE = \frac{n}{2}\ln(2\pi) + \frac{1}{2}\ln\left(\frac{\sigma^2}{1-\rho^2}\right)^n - \frac{1}{2}\rho^2\sigma^2\varepsilon_1^2 \tag{A-6.2}$$

$$- \frac{1}{2\sigma_v^2} \sum_{t=2}^{n} (\varepsilon_t - \rho\varepsilon_{t-1})^2$$

where

$$\sigma_v^2 = \frac{1}{n}\left[-\rho^2\varepsilon_1^2 + \sum_{t=2}^{n} (\varepsilon_t - \rho\varepsilon_{t-1})^2\right] \tag{A-6.3}$$

and ρ is estimated from the implicit equation:

$$\varepsilon_1^2 - \left(\sum_{t=2}^{n} \varepsilon_{t-1}^2\right)\rho^3 + \left(\sum_{t=2}^{n} \varepsilon_t^2\varepsilon_{t-1}^2\right) + \left(\sigma_v^2 - \varepsilon_1^2 + \sum_{t=2}^{n} \varepsilon_{t-1}^2\right)\rho \tag{A-6.4}$$

$$- \sum_{t=2}^{n} \varepsilon_t\varepsilon_{t-1} = 0$$

where

$\varepsilon_t = \tilde{z}_t - z_t(\underline{\theta})$;

ρ = first lag autocorrelation coefficient;

n = length of the data set; and

σ_v^2 = variance of the random portion of the errors.

The AMLE is solved in two stages. Given a set of model parameters, the residuals of the model ε_t are obtained. Next, equations (A-6.3) and (A-6.4) are solved to obtain estimates of σ_v^2 and ρ. These values are then substituted into Equation (A-6.2) to compute the value of the AMLE function.

A-6.3 The Maximum Likelihood Estimator for the Case of Gaussian Uncorrelated, Heteroscedastic Errors (HMLE)

The HMLE (from Sorooshian, Gupta, and Fulton, 1981) is the minimum variance asymptotically unbiased estimator when the errors are Gaussian, zero mean, uncorrelated, and have time-nonstationary variance. The variance of the errors is assumed to be related to the level of the output (magnitude of the flows). Such errors are believed to be common in streamflow data. The estimator is:

$$\underset{\text{wrt } \theta}{\text{Min}} \quad \text{HMLE} = \left[\sum_{t=1}^{n} w_t \varepsilon_t^2\right] / \left[\prod_{t=1}^{\nu} w_t\right]^{1/n} \tag{A-6.5}$$

where

$$w_t = f_t^{2(\lambda-1)} \tag{A-6.6}$$

and is estimated by solving the implicit equation:

$$\sum_{t=1}^{n} \ln(f_t) \cdot \sum_{t=1}^{n} w_t \varepsilon_t^2 - n \cdot \sum_{t=1}^{n} w_t \cdot \ln(f_t) \varepsilon_t^2 = 0 \tag{A-6.7}$$

where

$$f_t = E\{\tilde{z}_t\}$$

$$= \tilde{z}_t \text{ or } z_t(\underline{\theta}) \text{ as desired}$$

The choice of f_t is discussed in Sorooshian et al. (1981). In the above equations, the λ is the unknown variance stabilization parameter, which is estimated during the optimization process. $E\{\tilde{z}_t\}$ denotes the expectation operation. The HMLE is solved in two stages. Given a set of model parameters, the residuals of the model are obtained. Next, Equation (A-6.7) is solved to obtain an estimate of the parameter λ. This value is substituted into Equation (A-6.5) to compute the value of the HMLE function.

A complete discussion of the maximum likelihood estimators included here can be found in either Sorooshian (1978) or Sorooshian, Gupta, and Fulton (1981).

APPENDIX 7

DATA SETS USED

A-7.1 Precipitation Input RAIN-1

I	P(I)	I	P(I)	I	P(I)	I	P(I)	I	P(I)
1.	7.0	11.	35.0	21.	14.0	31.	4.0	41.	11.0
2.	12.0	12.	27.0	22.	8.0	32.	9.0	42.	6.0
3.	6.0	13.	12.0	23.	0.0	33.	3.0	43.	0.0
4.	4.0	14.	4.0	24.	0.0	34.	0.0	44.	0.0
5.	0.0	15.	0.0	25.	0.0	35.	0.0	45.	0.0
6.	0.0	16.	0.0	26.	0.0	36.	3.0	46.	2.0
7.	0.0	17.	0.0	27.	0.0	37.	14.0	47.	8.0
8.	0.0	18.	0.0	28.	0.0	38.	20.0	48.	3.0
9.	9.0	19.	3.0	29.	0.0	39.	37.0	49.	3.0
10.	17.0	20.	6.0	30.	0.0	40.	18.0	50.	0.0

A-7.2 Streamflow Output FLOW-1

I	Q(I)	I	Q(I)	I	Q(I)	I	Q(I)	I	Q(I)
1.	1.40	11.	27.62	21.	5.44	31.	1.54	41.	12.50
2.	3.52	12.	27.00	22.	6.83	32.	3.03	42.	9.25
3.	4.16	13.	13.00	23.	4.85	33.	3.02	43.	5.58
4.	4.49	14.	8.50	24.	3.69	34.	2.42	44.	4.21
5.	3.14	15.	5.35	25.	2.82	35.	1.94	45.	3.13
6.	2.46	16.	4.06	26.	2.25	36.	2.15	46.	2.87
7.	1.97	17.	3.04	27.	1.80	37.	4.00	47.	3.89
8.	1.57	18.	2.41	28.	1.44	38.	11.30	48.	4.05
9.	3.06	19.	2.53	29.	1.15	39.	34.30	49.	3.55
10.	6.62	20.	3.22	30.	0.92	40.	18.00	50.	2.79

A-7.3 Precipitation Input RAIN-4

I	P(I)	I	P(I)	I	P(I)	I	P(I)	I	P(I)
1.	5.0	11.	16.0	21.	6.0	31.	20.0	41.	17.0
2.	5.0	12.	20.0	22.	4.0	32.	27.0	42.	13.0
3.	7.0	13.	28.0	23.	0.0	33.	33.0	43.	10.0
4.	8.0	14.	26.0	24.	0.0	34.	37.0	44.	8.0
5.	10.0	15.	22.0	25.	2.0	35.	35.0	45.	5.0
6.	15.0	16.	18.0	26.	3.0	36.	33.0	46.	3.0
7.	14.0	17.	16.0	27.	5.0	37.	31.0	47.	2.0
8.	15.0	18.	12.0	28.	6.0	38.	29.0	48.	0.0
9.	17.0	19.	10.0	29.	12.0	39.	26.0	49.	0.0
10.	14.0	20.	10.0	30.	14.0	40.	21.0	50.	0.0

A-7.4 Streamflow Output FLOW-4

I	Q(I)	I	Q(I)	I	Q(I)	I	Q(I)	I	Q(I)
1.	1.00	11.	16.00	21.	8.38	31.	15.69	41.	17.00
2.	1.80	12.	20.00	22.	6.51	32.	27.00	42.	13.50
3.	2.84	13.	28.00	23.	4.75	33.	33.00	43.	11.75
4.	3.87	14.	26.00	24.	3.53	34.	37.00	44.	9.88
5.	5.85	15.	22.00	25.	3.12	35.	35.00	45.	7.44
6.	9.32	16.	18.00	26.	3.09	36.	33.00	46.	5.93
7.	11.66	17.	16.00	27.	3.47	37.	31.00	47.	5.17
8.	13.33	18.	13.00	28.	4.04	38.	29.00	48.	3.63
9.	16.33	19.	11.50	29.	6.48	39.	26.00	49.	2.74
10.	14.00	20.	10.75	30.	9.69	40.	21.00	50.	2.19

APPENDIX 8

PROCEDURE FOR CREATING ERROR CORRUPTED OUTPUT DATA

The following procedure was used to generate the error corrupted flow sequences used in the simulation experiments of Chapter 4. It is assumed that the actual measurements are made of river stage which are then converted to flow measurements through the nonlinear rating curve. The procedure outlined is for Gaussian, zero mean, stage measurement errors, with constant variance σ^2 and autocorrelated to single lag with autocorrelation coefficient Φ_1. In the studies conducted in Chapter 4, ϕ_1 was assumed to be zero, σ^2 was assumed to be 0.25, and γ was equal to either 1.0, 0.5, or 0.05.

Step 1

A sequence a_t of Gaussian, zero mean, unit variance pseudorandom numbers was generated according to the following method:

(1) Generate two independently distributed uniform pseudorandom numbers, ν_1 and ν_2, between zero and one using the built-in computer function.

(2) Compute two pseudorandom Gaussian (N[0,1]) numbers using:

$$\xi_1 = (-2\pi \ln\nu_1)^{1/2} \cdot \mathrm{Cos}(2\pi\nu_1) \qquad \text{(A-8.1)}$$

$$\xi_2 = (-2\pi \ln\nu_2)^{1/2} \cdot \mathrm{Sin}(2\pi\nu_2) \qquad \text{(A-8.2)}$$

(3) Repeat Steps 1 and 2 until enough numbers are generated.

(4) Test the mean, variance, and autocorrelation function of the sequence.

Step 2

The "observed" flow sequences were created using the following method (h_t = stage, z_t = flow):

(1) Compute $$h_{t,true} = \beta + (z_{t,true})^{\gamma}/\gamma \quad \text{(A-8.3)}$$

(2) Compute $$v_t = \phi_1 v_{t-1} + a_t \quad \text{(A-8.4)}$$

where $v_1 = a_1$

(3) Compute observed stage $\tilde{h}_t = h_{t,true} + v_t$ (A-8.5)

(4) If $\tilde{h}_t < 0.0$, set $\tilde{h}_t = 10^{-6}$

(5) Compute observed flows, $\tilde{z}_t = [\gamma(h_t - \beta)]^{1/\gamma}$ (A-8.6)

In this report, the value $\beta = 0.0$ was used.

The procedure for generating the random numbers was found to be satisfactory. The means, variances, and autocorrelations all lay within their 95% confidence intervals, the histograms appeared acceptable, and the cumulative probability distributions plotted as straight lines on normal probability paper.

APPENDIX 9

DERIVATION OF DERIVATIVES FOR CRR MODELS

APPENDIX 9

The model equations are given by:

$$\underline{x}_t = \sum_{j=1}^{q} T_j\,(\underline{\theta}, \underline{x}_{t-1}, \underline{u}_t) \cdot A_j\,(\underline{\theta}, \underline{x}_{t-1}, \underline{u}_t) \qquad \text{(A-9.1)}$$

$$\underline{z}_t = \sum_{j=1}^{q} T_j\,(\underline{\theta}, \underline{x}_{t-1}, \underline{u}_t) \cdot B_j\,(\underline{\theta}, \underline{x}_{t-1}, \underline{u}_t) \qquad \text{(A-9.2)}$$

Where $T_j(\cdot)$ is defined in equations (3.37) and (3.38). Let us denote $A_j\,(\underline{\theta}, \underline{x}_{t-1}, \underline{u}_t)$, $B_j\,(\underline{\theta}, \underline{x}_{t-1}, \underline{u}_t)$ and $T_j\,(\underline{\theta}, x_{t-1}, \underline{u}_t)$ by the short forms A_{jt}, B_{jt}, and T_{jt}, respectively. Noting that the model equations are continuous in the state and parameter spaces for each mode, we get the following expressions through the application of the chain rule of differentiation,

$$\frac{\partial \underline{X}_t}{\partial \theta_i} = \sum_{j=1}^{q} \left[T_{jt} \frac{\partial A_{jt}}{\partial \theta_i} + \frac{\partial T_{jt}}{\partial \theta_i} \cdot A_{jt} \right] \qquad \forall\ i \qquad \text{(A-9.3)}$$

$$\frac{\partial \underline{Z}_t}{\partial \theta_i} = \sum_{j=1}^{q} \left[T_{jt} \frac{\partial B_{jt}}{\partial \theta_i} + \frac{\partial T_{jt}}{\partial \theta_i} \cdot B_{jt} \right] \qquad \forall\ i \qquad \text{(A-9.4)}$$

where

$$\frac{\partial A_{jt}}{\partial \theta_i} = \frac{\partial A_{jt}}{\partial \theta_i} + \frac{\partial A_{jt}}{\partial \underline{X}_{t-1}} \cdot \frac{\partial \underline{X}_{t-1}}{\partial \theta_i} \qquad \forall\ i,j \qquad \text{(A-9.5)}$$

$$\frac{\partial B_{jt}}{\partial \theta_i} = \frac{\partial B_{jt}}{\partial \theta_i} + \frac{\partial B_{jt}}{\partial \underline{X}_{t-1}} \cdot \frac{\partial \underline{X}_{t-1}}{\partial \theta_i} \qquad \forall\ i,j \qquad \text{(A-9.6)}$$

$$\frac{\partial T_{jt}}{\partial \theta_i} = \frac{\partial T_{jt}}{\partial \theta_i} + \frac{\partial T_{jt}}{\partial \underline{X}_{t-1}} \cdot \frac{\partial \underline{X}_{t-1}}{\partial \theta_i} \qquad \forall\ i,j \qquad \text{(A-9.7)}$$

Substituting Equations (A-9.5) through (A-9.7) into (A-9.3) and (A-9.4), we get:

$$\frac{\partial \underline{X}_t}{\partial \theta_i} = \sum_{j=1}^{q} T_{jt} \left[\frac{\partial A_{jt}}{\partial \theta_i} + \frac{\partial A_{jt}}{\partial \underline{X}_{t-1}} \cdot \frac{\partial \underline{X}_{t-1}}{\partial \theta_i} \right]$$

$$+ \sum_{j=1}^{q} \left[\frac{\partial T_{jt}}{\partial \theta_i} + \frac{\partial T_{jt}}{\partial \underline{X}_{t-1}} \cdot \frac{\partial \underline{X}_{t-1}}{\partial \theta_i} \right] A_{jt} \qquad \forall\ i \qquad \text{(A-9.8)}$$

$$\frac{\partial \underline{Z}_t}{\partial \theta_i} = \sum_{j=1}^{q} T_{jt} \left[\frac{\partial B_{jt}}{\partial \theta_i} + \frac{\partial B_{jt}}{\partial \underline{X}_{t-1}} \cdot \frac{\partial \underline{X}_{t-1}}{\partial \theta_i} \right]$$

$$+ \sum_{j=1}^{q} \left[\frac{\partial T_{jt}}{\partial \theta_i} + \frac{\partial T_{jt}}{\partial \underline{X}_{t-1}} \cdot \frac{\partial \underline{X}_{t-1}}{\partial \theta_i} \right] B_{jt} \qquad \forall\ i \qquad \text{(A-9.9)}$$

Since each mode is continuous in $\underline{\theta}$ and $\underline{X}_{t-1}$, the quantities A_{jt}, B_{jt}, $\frac{\partial A_{jt}}{\partial \theta_i}$, $\frac{\partial A_{jt}}{\partial \underline{X}_{t-1}}$, $\frac{\partial B_{jt}}{\partial \theta_i}$ and $\frac{\partial B_{jt}}{\partial \underline{X}_{t-1}}$ can be easily computed. Further, the quantity $\frac{\partial \underline{X}_{t-1}}{\partial \theta_i}$ is the sensitivity of the model state to the parameter θ_i carried over from the previous time step and is therefore already known. In order to compute the above equations, we now need to compute T_{jt}, $\frac{\partial T_{jt}}{\partial \theta_i}$ and $\frac{\partial T_{jt}}{\partial \underline{X}_{t-1}}$. Note that T_{jt} is a piecewise constant function on the state and parameter spaces. For example, let the model consist of just a single state x_t, and two operational modes. Figure A-9.1 represents the T_{jt} function ($j = 1,2$) for the 2-mode model, as a function of the state. From the figure we can see that (for $j = 1,2$)

$$\frac{\partial T_{jt}}{\partial x_{t-1}} = \begin{cases} 0 & \text{for } x_{t-1} \neq a \\ \text{undefined} & \text{for } x_{t-1} = a \end{cases} \qquad \text{(A-9.10)}$$

where 'a' represents the value of the state x_{t-1} at which the model switches between the modes. As long as $x_{t-1} \neq a$, we know that the partial derivative

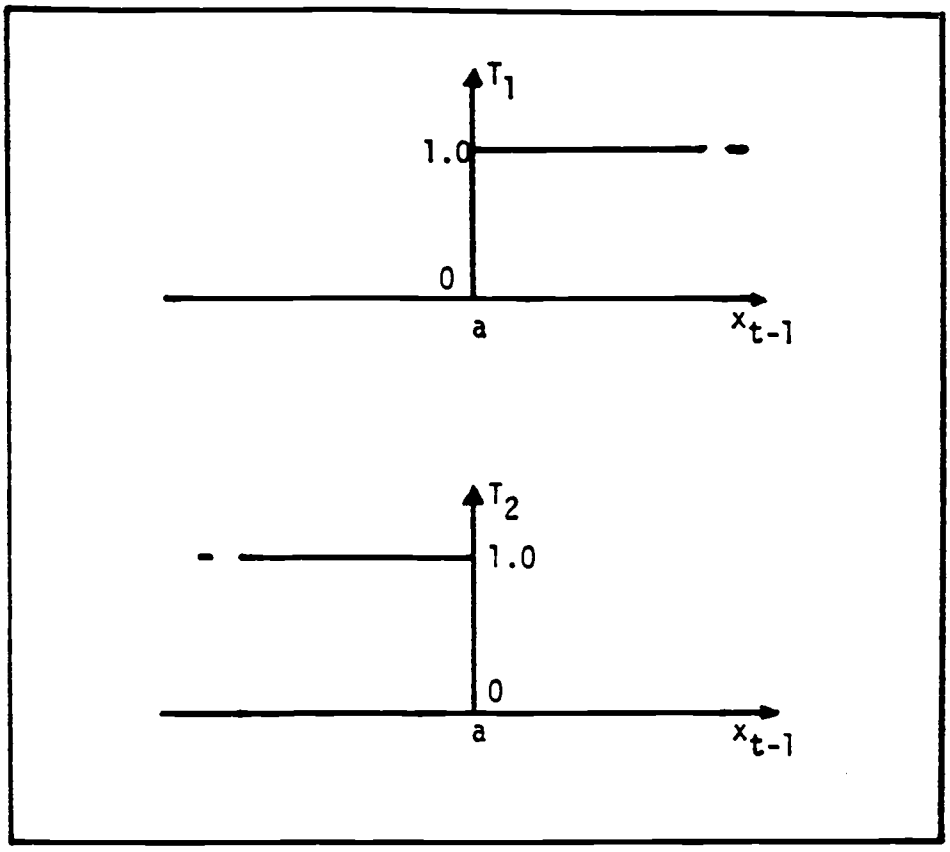


Figure A-9.1. Threshold function for a 2-mode model.

$\frac{\partial T_{jt}}{\partial x_{t-1}}$ is equal to zero. As we allow x_{t-1} to approach the value 'a' from each side, the one-sided limiting values of the derivative is zero, but is undefined at the point a. This problem is most simply resolved by defining the threshold function T_{jt} to be continuous from the left (or alternatively, from the right). For example, for the case of figure A-9.1:

$$T_{1t} = \begin{cases} 0 & x_{t-1} \leq a \\ 1 & x_{t-1} > a \end{cases} \qquad \text{(A-9.11)}$$

$$T_{2t} = \begin{cases} 1 & x_{t-1} \leq a \\ 0 & x_{t-1} > a \end{cases} \qquad \text{(A-9.12)}$$

which implies that

$$\frac{\partial T_{jt}}{\partial x_{t-1}} = \text{zero} \qquad \forall\ i,j \qquad \text{(A-9.13)}$$

In practice, a model which uses a threshold element is actually constructed using inequalities ($<$ or $>$) and is referred to implicitly as being continuous from one side or the other of the threshold point. The threshold point 'a' is rarely, if ever, treated as a singularity. It may be argued that this approach does not resolve the problems associated with nondifferentiability of the function at the point $x_{t-1} = a$. Based on our experience with application of the method in simulation studies, it appears, however, that the problems are probably only theoretical and not of much practical significance. A similar approach may be used to establish that $\frac{\partial T_{jt}}{\partial \theta_i} = 0$ (for all i,j) as a function of the parameter space.

These results are extended readily to the multistate, multimode case as follows. Let the model be in mode j when $\underline{a}_j < \underline{X}_{t-1} < \underline{b}_j$ where $\underline{a}_j$ and $\underline{b}_j$ denote the values of the state $\underline{X}_{t-1}$ at which the model changes from the jth mode of

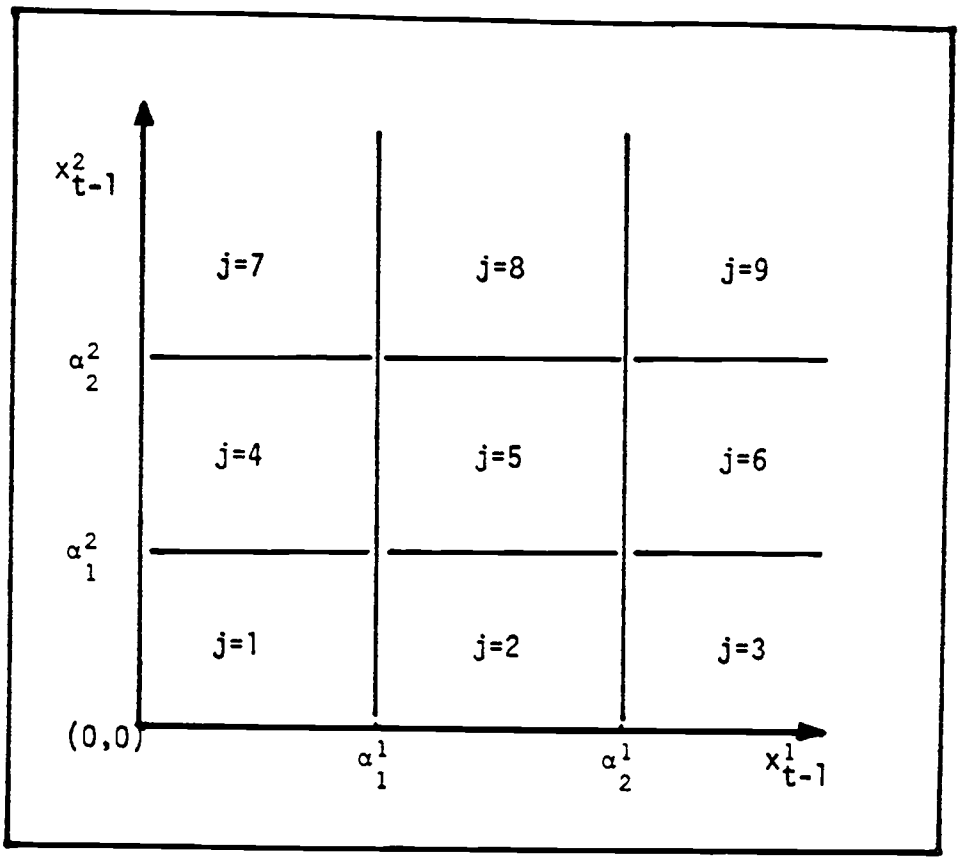


Figure A-9.2. Example of a two-state model with nine possible modes of operation.

(j) - indicates model in mode j (j = 1,...9).

REFERENCES

Aitken, A.P., 1973, "Assessing Systematic Errors in Rainfall-Runoff Models," Journal of Hydrology, 20(2), pp. 121-136.

Akaike, H., 1974, "A New Look at Statistical Model Identification," IEEE Transactions on Automatic Control, Vol. AC-19, No. 6, pp. 716-723.

Alley, W.M., D.R. Dawdy, and J.C. Schaake, Jr., 1980, "Parametric-Deterministic Urban Watershed," J. Hyd. Div. ASCE, 106 (HY5), pp. 679-690.

Arfi, F., 1980, Response Surface Methodology as a Parameter Sensitivity Analysis Tool for Rainfall-Runoff Models, M.S. Thesis, Dept. of Systems Engineering, Case Western Reserve University, Cleveland, Ohio.

Baram, Y., and N.R. Sandell, 1978, "An Information Theoretic Approach to Dynamical Systems Modeling and Identification," IEEE Transactions on Automatic Control, Vol. AC-23, No. 1, pp. 61-66.

Bard, Y., 1974, Non-Linear Parameter Estimation, Academic Press, New York.

Beard, L.R., 1967, "Optimization Techniques for Hydrologic Engineering," Water Resources Research, 3(3), pp. 809-815.

Bellman, R.E., and L.A. Zadeh, 1970, "Decision Making in a Fuzzy Environment," Management Science, 17B, pp. 141-163.

Bloomfield, P.H.I'a., D.H. Pilgrim, and K.K. Watson, 1981, "The Infiltration-Soil Water Storage Relationship at a Point in Watershed Modeling," Water Resources Research, 17(2), pp. 370-376.

Boughton, W.C., 1965, A New Simulation Technique for Estimating Catchment Yield, Rep. 78, Water Resources Lab, University of New South Wales, Manly Vale, Australia.

Boughton, W.C., 1966, "A Mathematical Model for Relating Runoff to Rainfall with Daily Data," Civil Eng. Trans. Inst. Eng. Aust., CE8(1), pp. 83-97.

Box, G.E.P., and G.M. Jenkins, 1976, Time Series Analysis: Forecasting and Control, Holden Day, San Francisco, California.

Brazil, L.E., and M.D. Hudlow, 1981, Calibration Procedures Used with the National Weather Service River Forecast System," in Y.Y. Haimes and J. Kindler (Eds.), Water and Related Land Resource Systems, IFAC Symp., Cleveland, Ohio, May 28-31, 1980, Pergamon, New York, pp. 457-466.

Burnash, R.J.C., K.L. Ferral, and R.A. McGuire, 1973, A Generalized Streamflow System--Conceptual Modeling for Digital Computers, Report, Joint Fed. State River Forecast Center, U.S. National Weather Service and California Dept. of Water Resources, Sacramento, California.

Carrigan, P.H., 1973, Calibration of U.S. Geological Survey Rainfall-Runoff Model for Peak Flow Synthesis--National Basins, U.S. Geological Survey Computer Report, U.S. Dept. of Commerce National Technical Information Service.

Chapman, T.G., 1970, Optimization of a Rainfall-Runoff Model for an Arid Zone Catchment, UNESCO Publication No. 96, Symposium of Wellington (New Zealand), pp. 126-143.

Claborn, B.J., and W. Moore, 1970, Numerical Simulation in Watershed Hydrology, Hydraulic Eng. Lab, University of Texas, Report No. HYD 14-7001.

Clarke, R.T., 1973, Mathematical Models in Hydrology, FAO Publication No. 19, Rome.

Crawford, N.H., 1971, Studies in the Application of Digital Simulation to Urban Hydrology, Hydrocomp International, Inc., Palo Alto, California.

Crawford, N.H., and R.K. Linsley, 1966, Digital simulation in hydrology: Stanford watershed model Mark IV, Dept. of Civil Engineering, Tech. Report 39, Stanford University, Stanford, California.

Cunday, T.W., and K.N. Brooks, 1981, "Calibrating and Verifying the SSARR Model: Missouri River Watersheds Study," Water Resources Bulletin, 17(5), pp. 775-782.

Dawdy, D.R., and T. O'Donnell, 1965, "Mathematical Models of Catchment Behavior," Journal of Hydraulics Division, ASCE, 91(HY4), pp. 113-137.

Dawdy, D.R., and R.W. Lichty, 1968, "Use of Analog and Digital Computers in Hydrology-Methodology of Hydrologic Model Building," Int. Assoc. Sci. Hydrol., Publ. 81, pp. 347-355.

Dracup, J.A., and T.J. Fogarty, et al. (names here?), 1973, Synthesis and Evaluation of Urban-Regional Hydrologic Rainfall-Runoff Criteria, Environmental Dynamic, Inc., Los Angeles, California.

Edwards, A.W.F., 1972, Likelihood, Cambridge University Press.

Eykhoff, P., 1982, "On the Coherence Among the Multitude of System Identification Methods," Proc. 6th IFAC Symposium on Identification and System Parameter Estimation, Washington, D.C., pp. 737-747.

Fiering, M.B., and G. Kuczera, 1982, "Robust Estimators in Hydrology," In: Scientific Basis of Water Resource Management, National Academy Press, pp. 85-94.

Fleming, G., 1975, Computer Simulation Techniques in Hydrology, Elsevier Scientific Publishing Company, New York.

Fulton, J.L., 1982, Discussion, Modification, and Extension of Some Maximum Likelihood Techniques for Model Calibration with Application to Rainfall-Runoff Models, M.S. Thesis, Dept. of Systems Engineering, Case Western Reserve University, Cleveland, Ohio.

Garrick, M., Cunnane, C., and Nash, J.E., 1978, "A Criterion of Efficiency for Rainfall-Runoff Models," Journal of Hydrology, 36(3/4), pp. 375-381.

Gertler, J., 1979, "A Constrained Minimum Variance Input-Output Estimator for Linear Dynamic Systems," Automatica, Vol. 15, pp. 353-358.

Goldstein, J.D., and W.E. Larimore, 1980, Applications of Kalman Filtering and Maximum Likelihood Parameter Identification to Hydrologic Forecasting, The Analytic Sciences Corporation, 6 Jacob Way, Reading, Massachusetts, Technical Report TR-1480-1.

Green, D.W., H. Dabri, C.F. Weinang, and R. Prill, 1970, "Numerical Modeling of Unsaturated Ground-water Flow and Comparison of the Model to a Field Experiment," Water Resources Research, 6(3), pp. 862-874.

Gupta, V.K., and S. Sorooshian, 1983, "Uniqueness and Observability of Conceptual Rainfall-Runoff Model Parameters: The Percolation Process Examined," Water Resources Research, 19(1), pp. 269-276.

Guttman, I., S.S. Wilks, J.S. Hunter, 1971, Introductory Engineering Statistics, 2nd Edition, John Wiley and Sons.

Haan, C.T., 1977, Statistical Methods in Hydrology, Iowa State University Press.

Hanks, R.J., A. Klute, and E. Brester, 1969, "A Numerical Method for Estimating Infiltration, Redistribution, Drainage, and Evaporation of Water from Soil," Water Resources Research, 5(5), pp. 1064-1069.

Holtan, H.N., and N.C. Lopez, 1973, USDAHL-73 Revised Model of Watershed Hydrology, U.S. Dept. of Agriculture, Plant Physiology Institute, Report No. 1.

Hooke, R., and T.A. Jeeves, 1961, "Direct Search Solutions of Numerical and Statistical Problems," J. ACM, 8(2), pp. 212-229.

Ibbitt, R.P., 1970, Systematic Parameter Fitting for Conceptual Models of Catchment Hydrology, Ph.D. Dissertation, University of London.

Ibbitt, R.P., and P.D. Hutchinson, 1983, "Model Parameter Consistency and Fitting Criteria," Report, Ministry of Works and Development, Box 1479, Christchurch, New Zealand.

Johnston, P.R., and D. Pilgrim, 1976, "Parameter Optimization for Watershed Models," Water Resources Research, 12(3), pp. 477-486.

Kailath, T., 1980, Linear Systems, Prentice-Hall, Inc., 682 p.

Karnopp, D.C., 1963, Random Search Technique for Optimization Problems, Automatica, Vol. 1, pp. 111-121, Pergamon Press.

Kashyap, R.L., 1982, "Optimal Choice of AR and MA Parts in Autoregressive Moving Average Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-4, No. 2, pp. 99-104.

Kibler, D.F., and Hipel, D.W., 1979, "Review of Geophysics and Space Physics: Surface Water Hydrology," Papers in Hydrology, U.S. National Report, 17th General Assembly, International Union of Geodesy and Geophysics, Canberra, Australia, December 2-15.

Kitanidis, P.K., and R.L. Bras, 1978, Real-Time Forecasting of River Flows, Ralph M. Parsons Lab for Water Resources and Hydrodynamics, Dept. of Civil Engineering, M.I.T., Technical Report.

Kitanidis, P.K., and R.L. Bras, 1980a, Adaptive filtering through detection of isolated transient errors in rainfall-runoff models, Water Resources Research, 16(4), 740-748.

Kitanidis, P.K., and R.L. Bras, 1980b, Real-time forecasting with a conceptual hydrologic model, 1, Analysis of uncertainty, Water Resources Research, 16(6), 1025-1033.

Klemes, V., 1982, "Empirical and Causal Models in Hydrology," In: Scientific Basis of Water Resource Management, National Academy Press, pp. 95-104.

Koopmans, T.C., and O. Reiersol, 1950, "The Identification of Structural Characteristics," Ann. Math. Stats., No. 21, pp. 165-181.

Kuczera, G., 1982, "On the Relationship Between the Reliability of Parameter Estimates and Hydrologic Time Series Data Used in Calibration," Water Resources Research, 18(1), pp. 146-154.

Larimore, W.E., 1981, "Applications of Maximum Likelihood Parameter Identification to Hydrologic Forecasting," Report #TR-1480-3, The Analytic Sciences Corporation, 1 Jacob Way, Reading, Massachusetts 01867, June 1.

Lemmer, H.R., and A.R. Rao, 1983, Critical Duration Analysis and Parameter Estimation in ILLUDAS, Water Resources Research Center, Tech. Report #153, Purdue University, West Lafayette, Indiana.

Linsley, R.K., and W.C. Ackermann, 1942, "A Method of Predicting the Runoff from Rainfall," Transactions ASCE, 107, pp. 825-835.

Linsley, R.K., M.A. Kohler, and J.L.H. Paulhus, 1975, Hydrology for Engineers, 2nd Edition, McGraw-Hill Publishing Co.

Liou, E. Y-S, 1970, OPSET: Program for Computerized Selection of Watershed Parameter Values for the Stanford Watershed Model, University of Kentucky Water Resources Institute, Report No. 34.

Mahmoud, M.S., and M.G. Singh, 1981, Large Scale Systems Modeling, Pergamon Press, Oxford England, 326 p.

Manley, R.E., 1978, "Simulation of Flows in Ungaged Basins," Hydrological Sciences Bulletin, 23(1), pp. 85-101.

McCuen, R.H., 1973, "Component Sensitivity: A Tool for the Analysis of Complex Water Resource Systems," Water Resources Research, 9(1), pp. 243-246.

Mein, R.G., and B.M. Brown, 1978, "Sensitivity of Optimized Parameters in Watershed Models," Water Resources Research, 14(2), pp. 299-303.

Monro, J.C., 1971, Direct Search Optimization in Mathematical Modeling and a Watershed Model Application, NOAA Technical Memorandum, NWS HYDRO-12.

Moore, R.J., and R.T. Clarke, 1981, "A Distribution Function Approach to Rainfall-Runoff Modeling," Water Resources Research, 17(5), pp. 1367-1382.

Murray, D.L., 1970, Boughton's Daily Rainfall-Runoff Model Modified for the Brenig Catchment, International Association of Scientific Hydrology Publ. No. 96, pp. 144-161.

Nash, J.E., and J.V. Sutcliffe et al. (names here?), 1970, "River Flow Forecasting Through Conceptual Models, Part I--A Discussion of Principles," Journal of Hydrology, 10(3), pp. 282-290.

Nathanson, M.H., and G.M. Saidel, 1982, Criteria for optimal experimental design: Application to ferrokinetics, Internal Report, Dept. of Biomedical Engineering, Case Western Reserve University, Cleveland, Ohio.

Nelder, J.A., and R. Mead, 1965, "A Simplex Method for Function Minimization," The Computer Journal, 7(4), pp. 308-313.

Otter, P.W., 1981, "Identification and Estimation of Discrete State-Vector Models with Stochastic Inputs," Automatica, 17(2), pp. 389-391.

Phillips, B., 1981, Second Order Response Surface Designs for Parameter Sensitivity Estimation, Internal Report, Dept. of Systems Engineering, Case Western Reserve University, Cleveland, Ohio.

Pickup, G., 1977, Testing the Efficiencies of Algorithms and Strategies for Automatic Calibration of Rainfall-Runoff Models, Hydrological Science Bulletin, 22(2), pp. 257-274.

Plinston, D.T., 1972, "Parameter Sensitivity and Interdependence in Hydrologic Models," In: J.N.R. Jeffers (Ed.), Math. Models in Ecology, Blackwell, Oxford, pp. 234-247.

Posada, P.J.R., and R.L. Bras, 1982, Automatic Parameter Estimation of a Large Conceptual Rainfall-Runoff Model: A Maximum Likelihood Approach, Report No. 267, Dept. of Civil Engineering, M.I.T., Cambridge, Massachusetts.

Ricca, V.T., 1972, The Ohio State University Version of the Stanford Streamflow Simulation Model, Part I--Technical Aspects, Ohio State University, Columbus, Ohio.

Rockwood, D.M., 1958, "Columbia Basin Streamflow Routing by Computer," Transactions ASCE, Waterways Harbours Division.

Rosenbrock, H.H., 1960, "An Automatic Method of Finding the Greatest or Least Value of a Function," The Computer Journal, 3(8), pp. 175-184.

Rothenberg, T.J., 1971, "Identification in Parametric Models," Econometrica, 39, pp. 577-591.

Soderstrom, T., 1981, "Identification of Stochastic Linear Systems in Presence of Input Noise," Automatica, 17(1), pp. 713-725.

Sorooshian, S., 1978, Consideration of Stochastic Properties in Parameter Estimation of Hydrologic Rainfall-Runoff Models, Ph.D. Dissertation, University of California, Los Angeles.

Sorooshian, S., 1981, "Comparison of Two Direct Search Algorithms Used in Calibration of Rainfall-Runoff Models," In: Y. Haimes and J. Knolles (Eds.), Water and Related Land Resource Systems, IFAC Symposium, Cleveland, Ohio, May 28-31, 1980, Pergamon Press, New York, pp. 477-485.

Sorooshian, S., 1983, "Surface Water Hydrology: On-Line Estimation," Review of Geophysics and Space Physics, 21(3), pp. 706-721.

Sorooshian, S., and F. Arfi, 1982, "Response Surface Sensitivity Analysis Methods for Postcalibration Studies," Water Resources Research, 18(5), pp. 1311-1598.

Sorooshian, S., and J.A. Dracup, 1980, "Stochastic Parameter Estimation Procedures for Hydrologic Rainfall-Runoff Models: Correlated and Heteroscedastic Error Cases," Water Resources Research, 16(2), pp. 430-442.

Sorooshian, S., and V.K. Gupta, 1983b, Automatic Calibration of Conceptual Rainfall-Runoff Models: The question of parameter observability and uniqueness, Water Resources Research, 19(1), 260-268.

Sorooshian, S., V.K. Gupta, and J.L. Fulton, 1981, "Modifications to the Automatic Calibration Procedures of the NWSRFS Soil Moisture Accounting Model: The CIT Version," Report to the Hydrologic Research Lab, U.S. National Weather Service, NOAA, U.S. Department of Commerce.

Sorooshian, S., Y.Y. Haimes, V.K. Gupta, J.L. Fulton, and F. Arfi, 1981, "Advanced Calibration Methodology for Hydrologic Rainfall-Runoff Simulation Models," Report to the Office of Water Research and Technology, U.S. Department of the Interior, Washington, D.C.

Sorooshian, S., V.K. Gupta, and J.L. Fulton, 1983, "Evaluation of Maximum Likelihood Evaluation Techniques for Conceptual Rainfall-Runoff Models: Influence of Calibration Data Variability and Length on Model Credibility," Water Resources Research, 19(1), pp. 251-159.

U.S. Army Corps of Engineers, 1973, HEC-1 Flood Hydrograph Package, Users and Programmers Manuals, HEC Program 723-X6-L2010.

Viessman, W., J.W. Knapp, G.L. Lewis, and T.E. Harbaugh (Late), 1977, Introduction to Hydrology, 2nd Edition, Harper and Row, New York.

Williams, B.J., and W.W-G Yeh, 1983, "Parameter Estimation in Rainfall-Runoff Models," Journal of Hydrology, 63:373-393.

Yeh, W.W-G, 1982, "State-of-the-Art: Parameter Estimation in Rainfall-Runoff Modeling," In: V.P. Singh (Ed.), Statistical Analysis of Rainfall and Runoff, Proc. of the Intl. Symp. on Rainfall-Runoff Modeling, Mississippi State, Mississippi, May 18-21, 1981, Littleton, Colorado, pp. 491-504.