PROCEDURE FOR ESTIMATING MODEL PARAMETERS OF A MATHEMATICAL MODEL

Prepared for

USDA FOREST SERVICE

ROCKY MOUNTAIN FOREST AND RANGE

EXPERIMENT STATION

Flagstaff, Arizona

Prepared by

D. B. Simons

Associate Dean for Engineering Research
and Professor of Civil Engineering

and

R. M. Li

Assistant Professor of Civil Engineering

Civil Engineering Department
Engineering Research Center
Fort Collins, Colorado 80523

April 1976

CER75-76DBS-RML22

AUTHORIZATION

This research is sponsored by the USDA Forest Service, Rocky Mountain Forest and Range Experiment Station and supported with matching funds from the Colorado State University Experiment Station. The investigations are being conducted in accordance with the Research Agreement No. 16-541-CA between the Rocky Mountain Forest and Range Experiment Station and Colorado State University. D. Ross Carder is the authorized project leader for the Rocky Mountain Forest and Range Experiment Station and Daryl B. Simons is the principal investigator for Colorado State University. The period of agreement is from May 1, 1975 to December 31, 1976.

In accordance with the project contract and the study plan, this report on the model calibration procedure is submitted.

TABLE OF CONTENTS

I. INTRODUCTION

1.1 General

The determination of model parameters is an important aspect in the mathematical modeling of system response. The performance of a model is very much dependent on the results of model calibration. A systematic and reliable method for estimating model parameters must precede practical applications of a model.

In the application of a mathematical model, the identification of model parameters is often dependent on an optimization scheme. The dependency on the optimization scheme may be reduced if the model is formulated according to the physical significance. For either a "black box" model or a simulation model considering physical significance, the calibration of a model is necessary when the model contains unknown parameters. The parameters of a "black box" model are not physically significant and hence, they are usually not predictable. While the ranges of parameters of a simulation model with physical significance are well imposed by physical conditions or measured data, the exact values of the parameters which produce correct model response are usually not available. Hence, the model calibration is generally inevitable for most of the modeling problems.

The simplest calibration technique is the trial and error method. Except for some models which contain parameters with very narrow searching ranges, the trial and error procedure is inefficient for most of the problems. An efficient procedure is apparently needed for the model calibration.

1.2 Review of Model Calibration Techniques

There are many optimization techniques available for the purpose of model calibration. However, the usefulness of a particular optimization technique is very much dependent on the formulation of the model being calibrated.

Before reviewing the methods, it is necessary to define the standard model calibration problem in a mathematical form. This problem is

$$\underset{X_1,X_2,\ldots,X_{N_p}}{\text{Minimize}} \quad F(X_1,X_2,\ldots,X_{N_p}) \tag{1}$$

Subject to

$$X_i^{\ell} \leq X_i \leq X_i^{u} \quad \text{for} \quad i = 1,2,\ldots,N_p$$

in which N_p is the number of unknown parameters in a model, X_i's$(i=1,2,\ldots,N_p)$ are the unknown parameters, $F(X_1,X_2,\ldots,X_{N_p})$ is the objective function which is a function of $X_1,X_2,\ldots,X_{N_p}$ parameters, and X_i^{ℓ} and X_i^{u} are respectively the lower and the upper limits of the ith parameter. Usually the constrained regions $(X_i^{\ell} \leq X_i \leq X_i^{u})$ are much larger than the searched regions, thus, the constraints are not active. In this case, the problem may be simplified as an unconstrained minimization problem (Himmelblau, 1972).

The optimization function F is usually defined as the sum of the squares of deviations between the simulated and the measured response.

The available optimization techniques for model calibration can be categorized into the following seven methods.

Least Square Method. This is a very common technique and is only useful when F is of a quadratic and of explicit form. Overton (1968) approximated a unit hydrograph by a Fourier series having seven components and formulated F to be a quadratic and explicit equation. Applying the method of least square, he estimated a set of optimum parameters utilizing analytical solutions.

Univariate Search. This search method, intuitively the simplest, seeks the optimum value of F by changing only one of the parameter values at a time until the line optimum for that parameter is found. This results in search directions that are always parallel to the orthogonal coordinate axes. When all N_p parameter directions have been searched successively, a cycle is complete, and the search pattern is repeated starting with the best values of the X_i found so far. Beard (1967) presented a more sophisticated version of this method by gradually reducing the number of the X_i values that are changed during any one cycle, only those parameters that have the greatest effect on F being changed. The major weakness of this simple procedure is that it cannot optimize satisfactorily on problems where the response surface contours form a ridge structure inclined to the parameter axes. Such formations are common whenever there is some degree of dependence between parameters.

Rotating Coordinate Search. This search technique is often called Rosenbrock's (1960) method. The first cycle of this method is the same as for the univariate search. However, instead of continually searching the coordinates corresponding to the directions of the independent variables, an improvement is made after one cycle of the coordinate search by lining the search directions up into an orthogonal system, with the overall step on the previous stage as the first building block for the new search coordinates. This method rapidly lines up along a ridge, avoiding the weakness of the univariate search method. Ibbitt and O'Donnell (1971) concluded that Rosenbrock's method (1960) is the most effective of the nine methods they used for fitting the hydrologic catchment model described by Dawdy and O'Donnell (1965).

Conjugate Direction Search. This technique (Powell (1964), Zangwill (1967)), although applicable to nonquadratic objective functions, was developed to find the optimum of quadratic functions in a finite number of steps. This method utilizes a property of ellipses that the direction through the tangent points of two parallel lines and two concentric ellipses passes through the center of the elliptical system. The limitation in using this technique is that F must be an explicit and differentiable function, which is generally not true for model calibration problems.

Gradient Search Method. This method is also called the method of steepest descent. The search begins by calculating the partial derivatives of F with respect to each component X_i at some initial point. (For a nondifferentiable function, the partial derivatives can be approximated by a numerical method.) The vector of these derivatives is the gradient direction vector which represents the direction of maximum instantaneous rate of the gradient and it gives the direction for optimization but not the magnitude of the step size to take. The optimum step size in that direction can be determined by any effective one-dimensional search technique (see Himmelblau, 1972). Recently, Tuffuor and Labadie (1974) applied this technique to calibrate a rainfall-runoff model. This technique is applicable whenever the dimension of the optimization problem is small and the partial derivatives can be easily evaluated.

Quasilinearization Method. Quasilinearization is a technique that facilitates the reverse solution of a system of differential equations. It involves decoupling the system of differential equations by linearization into a series of initial value problems that may be repetitively solved in such a way that their solution converges to the solution of the original problem. Labadie and Dracup (1969) utilized this technique to estimate the parameters of a lumped watershed model. Yeh and Tauxe (1971) also successfully used this technique to calibrate an aquifer simulation model. As reported by Tuffuor and Labadie (1974) that the primary disadvantage of quasilinearization is its instability in solutions whenever a poor initial guess is chosen.

OPSET Method. OPSET program was developed by Liou (1970) for computerized selection of watershed parameter values for the Stanford Watershed Model. Liou (1970) reported that standard optimization techniques proved infeasible and other methods, which were based on the results of parameter sensitivity studies, were used. Basically, this program uses measurable watershed characteristics, climatological data and measured streamflow data to find the optimum set of parameters which define the various flow and storage functions. The optimization is done in two phases, a rough phase which uses large time increments, and a phase in which the results are refined by using finer time increments. This is because the rough phase may provide a very good initial approximation without requiring too much computer time.

The objective function in the parameter identification problem is generally not differentiable with respect to the parameters. This is due to the reason that the function is complicated with mathematical expressions and usually cannot be represented by a single equation. As the function is not differentiable, the optimization schemes using derivatives cannot be applied. An algorithm without using derivatives is often necessary for the calibration of a mathematical model.

In this study Powell's unidimensional minimization technique (Powell, 1964) is modified for use in calibrating the model with only one unknown parameter. The modifications on this technique have improved its efficiency. In addition, the Rosenbrock's (1960) optimization scheme is modified by coupling this modified Powell's unidimensional search technique to calibrate the model having multiple unknown parameters. The Rosenbrock's (1960) optimization technique is used because it is by far the most promising and efficient method for fitting a hydrologic model (Ibbitt and O'Donnell, 1971) and it also does not use derivatives of functions.

II. ONE-DIMENSIONAL CALIBRATION TECHNIQUE

2.1 Description of Method

The one-dimensional search technique is a fundamental component of any multidimensional search technique. A good unidimensional search technique is necessary not only for solving one-dimensional problems but also for improving multidimensional search techniques.

There are various methods for unidimensional searches. For example, uniform search, dichotomous search, Fibonacci search, Golden Section search, DSC unidimensional search and Powell's unidimensional minimization (Himmelblau, 1972). After a survey of these available methods, a method modified from Powell's unidimensional minimization method is developed in this study. The major modifications are to consider the convexity of the objective function and to allow constrained minimization problems.

For the one-dimensional problem, the functional representation is

$$\underset{X}{\text{Minimize}} \quad F\,(X)$$

Subject to (2)

$$X_\ell \leq X \leq X_u$$

in which X is the unknown parameter, and X_ℓ and X_u are respectively the lower and the upper limits of this parameter.

The method developed in this study is carried out using the first three points obtained in the direction of search. The X corresponding to the minimum of the quadratic function is determined, and these quadratic approximations are continued until the minimum of $F\,(X)$ is located to the required precision. The steps of the search are as follows (see Fig. 1):

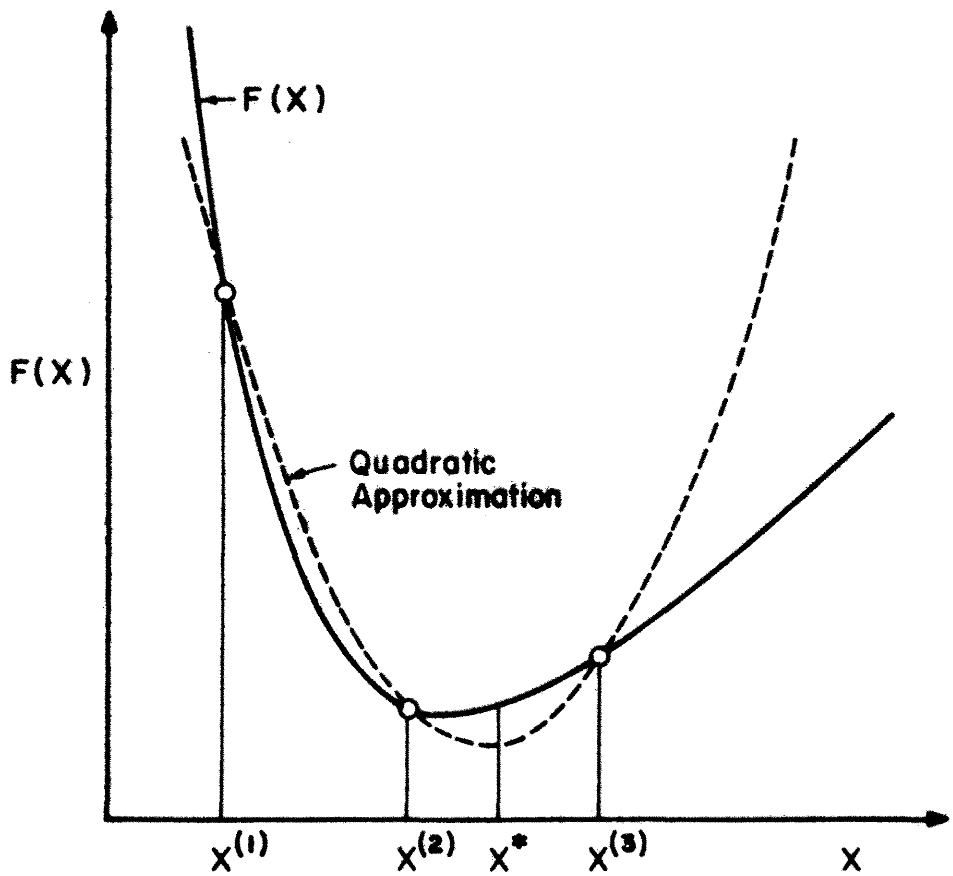

Fig. 1 Quadratic approximation for unidimensional search

<u>Step 1</u>. From the base vector $X^{(1)}$ compute

$$X^{(2)} = X^{(1)} + \Delta X \tag{3}$$

<u>Step 2</u>. Compute $F(X^{(1)})$ and $F(X^{(2)})$

<u>Step 3</u>. Determine the third point required for quadratic approximation.

When $F(X^{(1)})$ is greater than $F(X^{(2)})$, let

$$X^{(3)} = X^{(1)} + 2\Delta X \quad \text{if} \quad X^{(1)} + 2\Delta X \leq X_u \tag{4}$$

and

$$X^{(3)} = X_u \quad \text{if} \quad X^{(1)} + 2\Delta X > X_u \tag{5}$$

When $F(X^{(1)})$ is less than or equal to $F(X^{(2)})$, let

$$X^{(3)} = X^{(1)} - \Delta X \quad \text{if} \quad X^{(1)} - \Delta X \geq X_\ell \tag{6}$$

and

$$X^{(3)} = X_\ell \quad \text{if} \quad X^{(1)} - \Delta X < X_\ell \tag{7}$$

<u>Step 4</u>. Compute $F(X^{(3)})$.

<u>Step 5</u>. Check the convexity of the quadratic equation, the optimal coefficient a^* can be determined by

$$a^* = \frac{(X^{(2)}-X^{(3)})\,F(X^{(1)}) + (X^{(3)}-X^{(1)})\,F(X^{(2)}) + (X^{(1)}-X^{(2)})\,F(X^{(3)})}{(X^{(1)}-X^{(2)})\,(X^{(2)}-X^{(3)})\,(X^{(1)}-X^{(3)})} \tag{8}$$

If $a^* \geq 0$ the function is convex and the search is continued at step 6.

If $a^* < 0$ the function is concave, let

$$X_a = \text{Min} \ \{X^{(1)}, X^{(2)}, X^{(3)}\} \tag{9}$$

$$X_b = \text{Max} \ \{X^{(1)}, X^{(2)}, X^{(3)}\} \tag{10}$$

and return to step 3 and resume the search with the following information

$$\Delta X = X_b - X_a \tag{11}$$

$$X^{(1)} = X_a \tag{12}$$

$$F\ (X^{(1)}) = F\ (X_a) \tag{13}$$

$$X^{(2)} = X_b \tag{14}$$

$$F\ (X^{(2)}) = F\ (X_b) \tag{15}$$

<u>Step 6</u>. Estimate the value of X at the minimum of $F\ (X)$, X^*. Compute the other optimal coefficient using

$$b^* = \frac{F\ (X^{(1)}) - F\ (X^{(2)})}{X^{(1)} - X^{(2)}} - a^*\ (X^{(1)} + X^{(2)}) \tag{16}$$

Then, estimate X^* by

$$X^* = -\frac{b^*}{2a^*} \tag{17}$$

If $X_\ell \leq X^* \leq X_u$, the constraints are satisfied and the search is continued at step 7.

If $X^* > X_u$ or $X^* < X_\ell$, the constraint is violated and the boundary point is used as optimum value of X, i.e.,

$$X^* = X_u \quad \text{if} \quad F(X_a) > F(X_b) \tag{18}$$

and

$$X^* = X_\ell \quad \text{if} \quad F(X_a) \leq F(X_b) \tag{19}$$

Step 7. Compute $F(X^*)$.

Step 8. Termination of the search

Let X° = whichever of $\{X^{(1)}, X^{(2)}, X^{(3)}\}$ corresponds to the smallest $F(X)$. The termination of search is made if

$$\left| 1 - \frac{F(X^*)}{F(X^\circ)} \right| \leq \varepsilon \tag{20}$$

in which ε is the convergence tolerance. If the convergence criterion is not satisfied, the search is repeated returning to step 3 with the following information.

Let

$$X_a = \text{Min.} \ \{X^o, X^*\} \tag{21}$$

$$X_b = \text{Max} \ \{X^o, X^*\} \tag{22}$$

$$\Delta X = X_b - X_a \tag{23}$$

$$X^{(1)} = X_a \tag{24}$$

$$F(X^{(1)}) = F(X_a) \tag{25}$$

$$X^{(2)} = X_b \tag{26}$$

$$F(X^{(2)}) = F(X_b) \tag{27}$$

A computer program was developed to perform the above procedures. The listing of the computer program is given in Appendix A (PROGRAM UNIMO) and the flow chart is given in Fig. 2. The computer program is written in FORTRAN IV extended and has been tested on the CDC 6400 Computer at Colorado State University.

2.2 Instruction for Use

A detailed description of the input and output of the program is given herein. However, the input and output information required to evaluate the objective function are not given because they vary with models to be calibrated.

2.2.1 Input Data

The input to the program includes the title of the problem, the maximum limit of number of stage search, the numerical identification for controlling the output, the initial estimate of the vector, the initial step size of the search, the upper bound of the vector, the lower bound of the vector, and the convergence tolerance. There are only two input cards. They are described.

(a) Title Card, One card with Format (20A4)

Column	Mnemonic Name	Description
1-80	TITLE	Heading of the problem, which may consist of any alphabetical characters or numbers of 80 words.

(b) Information Card, One card with Format (2I10, 4F10,5 E103)

Column	Mnemonic Name	Description
1-10	MST	Maximum limit of number of stage search (number of quadratic approximation).
11-20	IPT	Numerical identification for controlling output information. = 0, only the final answer is printed.

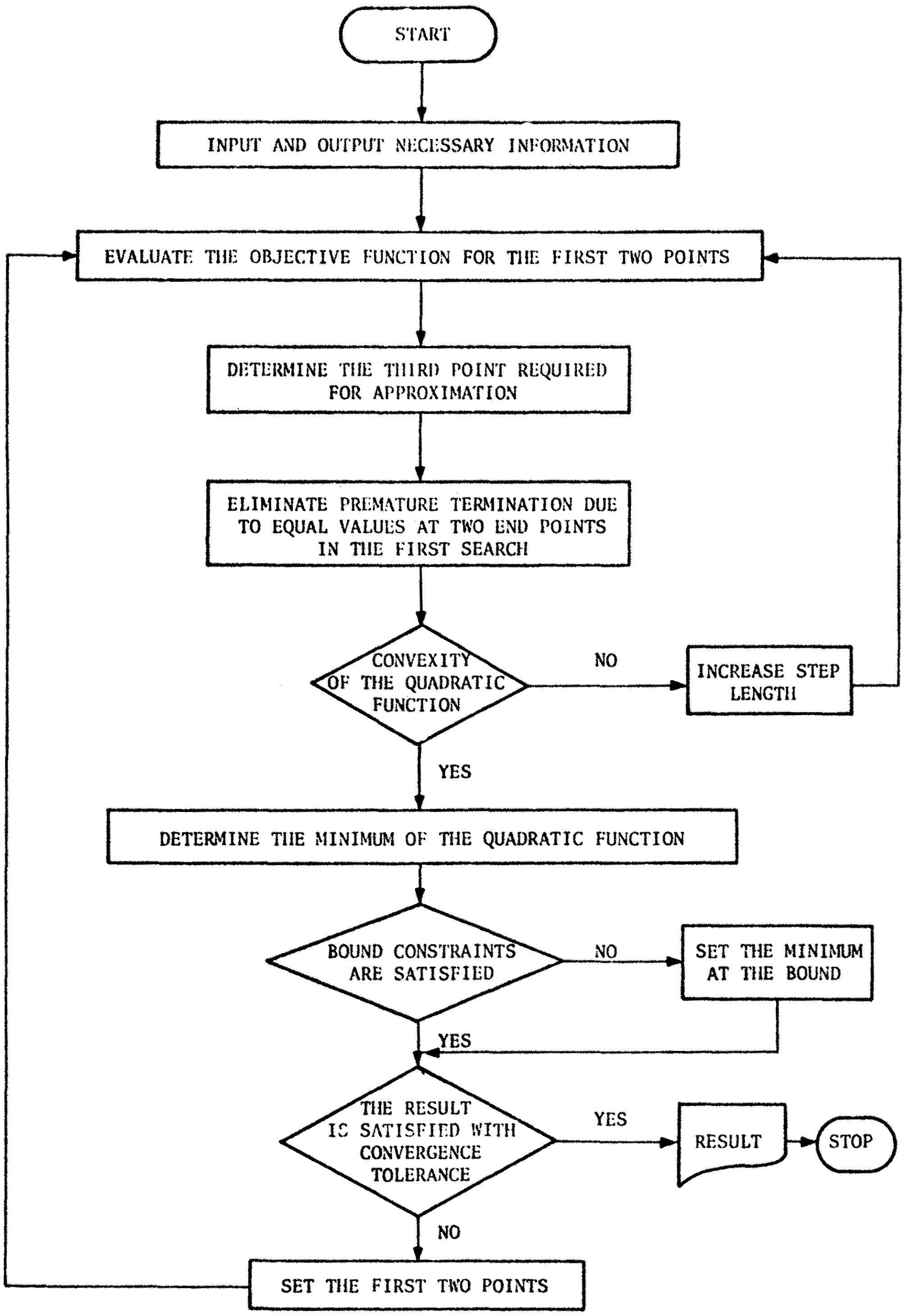

Fig. 2 FLOW CHART OF PROGRAM UNIMO

Column	Mnemonic Name	Description
		= 1, intermediate values of each stage search are printed
21-30	XA	Initial estimate of the vector
31-40	DX	Intial step size of search
41-50	XUPL	Upper limit of the vector
51-60	XLOL	Lower limit of the vector
61-70	EPS	Convergence tolerance

2.2.2 Output Information

The output from this computer program includes (1) all input data, (2) number of stage search, (3) intermediate values at the end of each stage search, (4) number of function evaluation, (5) optimum value of the objective function, and (6) optimum estimate of the vector. The Fortran labels of key output are listed below.

Mnemonic Name	Description
NS	Number of stage search or number of quadratic approximation
NEF	Number of function evaluation of the objective function
FSTA	Optimum value of the objective function
XSTA	Optimum estimate of the vector

2.3 Example

For simplicity a simple function is used as an example to demonstrate the application of the method.

In Fig. 3 the path of the search for the minimization of the following function by PROGRAM UNIMO is given.

$$F\ (X) = (1 - X^2)^2 + (1 - X)^2 \tag{28}$$

Equation 28 is often called the "Rosenbrock" function

The initial estimate of the vector $X^{(1)}$ is -2.0, the upper limit is 10.0, the lower limit is -10.0, the convergence tolerance, ε, is 1.0×10^{-3} and the initial step size of search, ΔX, is 0.5. The calibration results are: $X^* = 1.0$, $F\ (X^*) = 2.5 \times 10^{-28}$ and the number of function evaluation is 30.

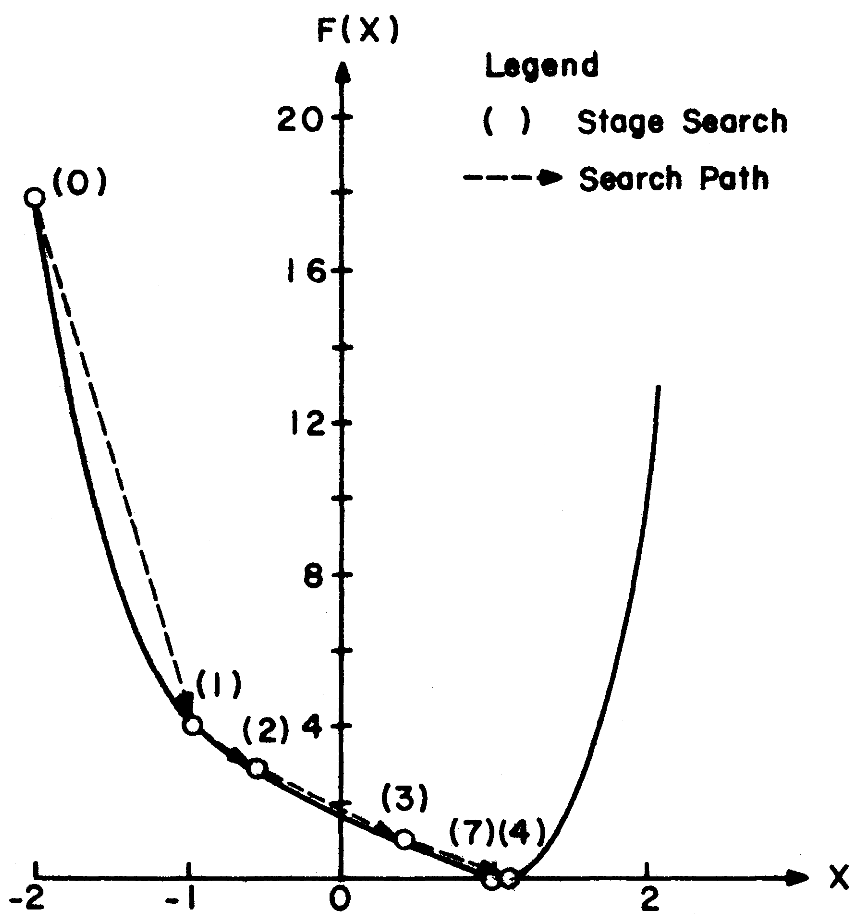

Fig. 3 Search path for the sample problem

III. MULTI-DIMENSIONAL CALIBRATION TECHNIQUE

3.1 Description of Method

Rosenbrock's method (1960) is an iterative procedure in which small steps are taken during the search in orthogonal coordinates. Instead of continually searching the coordinates corresponding to the directions of the independent variables, an improvement of search is made after one cycle of coordinate search by lining the search directions up into an orthogonal system, with the overall step of the previous stage as the first building block for the new search coordinates. Rosenbrock (1960) used an unconstrained dichotomous search to determine the search along a direction and generated the orthonormal set of directions by Gram-Schmidt procedure (Himmelblau, 1972).

In this study the Rosenbrock's optimization scheme (Rosenbrock, 1960) is modified by coupling the unidimensional search technique presented in Section II and by considering constrained minimization problems. In addition, Palmer's method (Palmer, 1969) for generating a new set of orthonormal search directions is used.

Let Y be a vector of $[X_1, X_2, \ldots, X_{N_p}]$. The method developed in this study locates the vector Y of the $(k+1)$-th stage by $Y^{(k+1)}$ by successive unidimensional searches from the vector Y of the k-th stage $Y^{(k)}$ along a set of orthonormal directions $\hat{S}_1^{(k)}$, $\hat{S}_2^{(k)}$, $\ldots$, $\hat{S}_{N_p}^{(k)}$. For the initial stage, $k = 0$, the directions $\hat{S}_1^{(0)}$, $\hat{S}_2^{(0)}$, $\ldots$, $\hat{S}_{N_p}^{(0)}$ are taken to be parallel to the axes of $X_1, X_2, \ldots, X_{N_p}$. More specifically let $Y_i^{(k)}$ indicate that the point at which $F\ (Y_i^{(k)}$ is a minimum in the direction of $\hat{S}_i^{(k)}$, for each stage (k) there are N_p vectors $Y_i^{(k)}$ and N_p optimal values of the objective function $F\ (Y_i^{(k)})$. From the initial vector $Y_0^{(k)}$, determine optimal step length $\lambda_1^{*(k)}$ in the direction of $\hat{S}_1^{(k)}$ so that $F\,(Y_0^{(k)} + \lambda_1^{*(k)}\,\hat{S}_1^{(k)})$ is a minimum and let $Y_1^{(k)} = Y_0^{(k)} + \lambda_1^{*(k)}\,\hat{S}_1^{(k)}$. Then from $Y_1^{(k)}$, determine $\lambda_2^{*(k)}$ so that $F(Y_1^{(k)} + \lambda_2^{*(k)}\,\hat{S}_2^{(k)})$ is a minimum and let $Y_2^{(k)} = Y_1^{(k)} + \lambda_2^{*(k)}\,\hat{S}_2^{(k)}$. The search pattern is generalized as follows; from $Y_{i-1}^{(k)}$, determine $\lambda_i^{*(k)}$ in the direction of $\hat{S}_i^{(k)}$ so that $F\,(Y_{i-1}^{(k)} + \lambda_1^{*(k)}\,\hat{S}_i^{(k)})$ is a minimum and let $Y_i^{(k)} = Y_{i-1}^{(k)} + \lambda_i^{*(k)}\,\hat{S}_i^{(k)}$. The search is repeated sequentially, always starting from the last immediate point in the sequence until all Y_i, $i=1, \ldots, N_p$ are determined. The unidimensional search technique developed in Section II is used to determine the optimal step length $\lambda_i^{*(k)}$. This constrained unidimensional search technique makes the multi-dimensional search method applicable in the constrained minimization problem described in Eq. 1.

After the kth stage has been completed, the vectors for the new search directions are computed at the point $Y_0^{(k+1)} = Y_{N_p}^{(k)}$. Palmer's method (Palmer, 1969), for generating a new set of search direction is used in this study. His method is as follows.

$$A_i^{(k)} = \sum_{j=i}^{N_p} \lambda_j^{*(k)}\,\hat{S}_j^{(k)} \quad \text{for} \quad 1 \leq i \leq N_p \tag{29}$$

in which $A_1^{(k)}$ is the vector from $Y_0^{(k)}$ to $Y_0^{(k+1)}$, $A_2^{(k)}$ is the vector from $Y_1^{(k)}$ to $Y_0^{(k+1)}$ and so on. $A_1^{(k)}$ represents the overall move from stage k to stage $(k+1)$, $A_2^{(k)}$ represents the overall move less the progress made during the search in direction $S_1^{(k)}$, etc. Then

$$\hat{S}_i^{(k+1)} = \frac{A_i^{(k)}\,||\,A_{i-1}^{(k)}\,||^2 - A_{i-1}^{(k)}\,||\,A_i^{(k)}\,||^2}{||\,A_{i-1}^{(k)}\,||\;||\,A_i^{(k)}\,||\,\sqrt{||\,A_{i-1}^{(k)}\,||^2 - ||\,A_i^{(k)}\,||^2}} \tag{30}$$

for $2 \leq i \leq N_p$

in which $||\;||$ is the norm of the vector and

$$S_1^{(k+1)} = \frac{A_1^{(k)}}{||\ A_1^{(k)}\ ||} \tag{31}$$

If $\lambda_{i-1}^{*(k)} = 0$, $\hat{S}_i^{(k+1)} = \hat{S}_{i-1}^{(k)}$ unless $\Sigma\ \lambda_i^{*(k)} = 0$. The search is terminated when

$$1 - \frac{F(Y_{N_p}^{(k+1)})}{F(Y_{N_p})^{(k)}} \leq \varepsilon \tag{32}$$

A computer program was developed to carry out the above procedure. In this program, the vector is normalized so that the ranges of the vector are within 0.0 and 1.0. The listing of the computer program is given in Appendix B. (PROGRAM BROSEN). Figure 4 gives a flow chart of the program. The computer program is written in FORTRAN IV EXTENDED and has been tested on CDC 6400 Computer at Colorado State University.

3.2 Instruction for Use

Presented in the following is a detailed description of the input and output information of the program. The input and output requirement for the objective function are not given because they are varied with models to be calibrated.

3.2.1 <u>Input Data</u>

The input to the program includes title of the problem, number of variables (or parameters), maximum limit of number of stage search, maximum limit of number of cycle search (number of stage search for unidimensional search), numerical identification for controlling output, convergence tolerance, initial estimate of the vector, initial step sizes of search, upper and lower bounds of the vector. There are three types of input cards which are described as follows.

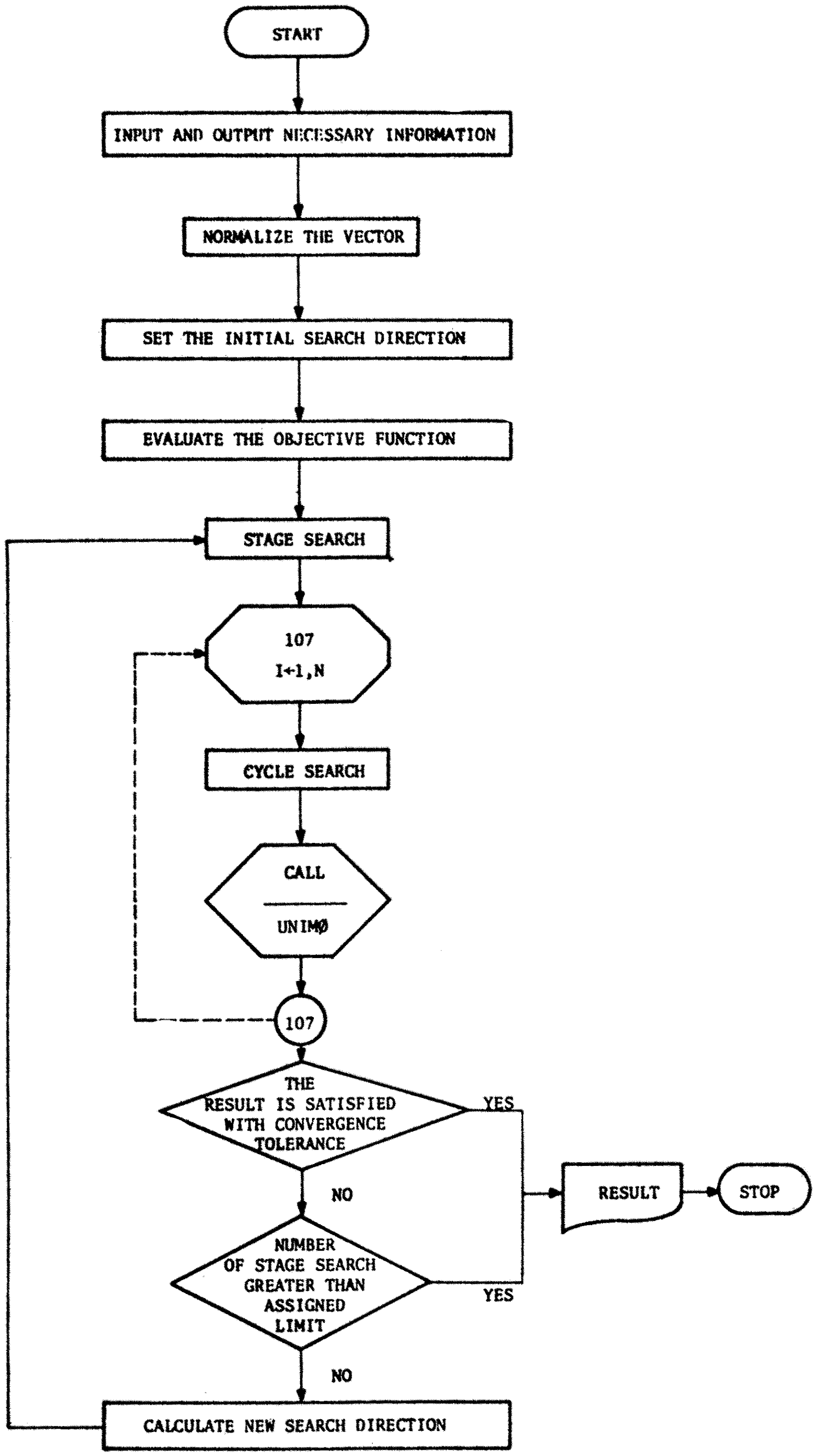

Fig. 4 FLOW CHART OF PROGRAM BROSEN

(a) Title Card, One card with Format (20A4)

Column	Mnemonic Name	Description
1-80	TITLE	Heading of the problem, which may consist of any alphabetical characters or number of 80 words.

(b) Information Card, One card with Format (4I10, E10.3)

Column	Mnemonic Name	Description
1-10	N	Number of variables (or parameters)
11-20	MST	Maximum limit of number of stage search (number of changing orthonormal directions)
21-30	MCL	Maximum limit of number of cycle search (number of quadratic approximation in the unidimensional search)
31-40	IPT	Numerical identification for controlling output information = 0, only the final answer is printed = 1, intermediate values of each stage search are printed = 2, intermediate value of each stage and cycle search are printed
41-50	EPS	convergence tolerance

(c) Vector Card, One card with Format (4F10.5) for every variable

Column	Mnemonic Name	Description
1-10	V(I)	Initial estimate of the I-th variable of the vector
11-20	D(I)	Initial step size of search along I-th search direction

Column	Mnemonic Name	Description
21-30	VUP(I)	Upper limit of the I-th variable of the vector
31-40	VLO(I)	Lower limit of the I-th variable of the vector

3.2.2 Output Information

The output from this computer program includes (1) all input data, (2) number of stage search, (3) intermediate values at the end of each stage and cycle search, (4) number of function evaluation, (5) optimum value of the objective function, and (6) optimum estimate of the vector. The Fortran labels of key output are listed below.

Mnemonic Name	Description
NS	Number of stage search or number of changing search directions
NEF	Number of function evaluation of the objective function
PO	Optimum value of the objective function
V(I)	Optimum estimate of the vector

3.3 Example

The number of function evaluations for the Rosenbrock's function (Rosenbrock, 1960) by the proposed algorithm is 30, which is much less than 206 function evaluations by the original Rosenbrock's method (Himmelblau, 1972). A sample problem with three variables is given herein for illustration.

The function is defined as

$$F\ (X) = (X_1 - X_2)^2 + (X_2 - 2X_3)^2 + (X_3 - 2)^2 \qquad (33)$$

This function is highly interactive among variables which is common for model calibration problems.

The initial estimate of the vector is

$$Y_0^{(0)} = [5.0,\ 2.0,\ 7.0] \tag{34}$$

The upper bound of the vector is

$$Y_u = [10.0,\ 10.0,\ 10.0] \tag{35}$$

The lower bound of the vector is

$$Y_\ell = [-10.0,\ -10.0,\ -10.0] \tag{36}$$

The convergence limit, $\varepsilon = 10^{-3}$

The search paths for each stage are given in Table 1. This table shows the applicability of the proposed algorithm for the problem with highly interactive parameters.

Table 1. Summary of Search Path for Each Stage of a Multi-dimensional Search Problem

Stage	Current Vector X_1	X_2	X_3	Current Objective Function	Cumulative No. Function Evaluation
0	5.000	2.000	7.000	0.178×10^{3}	0
1	2.000	8.000	3.600	0.392×10^{2}	16
2	7.005	8.220	3.386	0.549×10^{1}	29
3	8.142	7.709	3.435	0.295×10^{1}	42
4	7.871	7.366	3.336	0.252×10^{1}	53
5	5.950	5.847	2.751	0.694×10^{0}	68
6	4.213	4.278	2.112	0.198×10^{-1}	81
7	4.004	4.005	2.002	0.628×10^{-5}	96
8	4.000	4.000	2.000	0.101×10^{-7}	114
9	4.000	4.000	2.000	0.685×10^{-9}	130
10	4.000	4.000	2.000	0.378×10^{-9}	145
11	4.000	4.000	2.000	0.352×10^{-9}	158
12	4.000	4.000	2.000	0.309×10^{-9}	172
13	4.000	4.000	2.000	0.222×10^{-9}	185
14	4.000	4.000	2.000	0.385×10^{-10}	201
15	4.000	4.000	2.000	0.376×10^{-10}	215
16	4.000	4.000	2.000	0.376×10^{-10}	227

IV. APPLICATION STRATEGY

4.1 General

The purpose of calibrating a mathematical model is to find a set of model parameters which produce correct system response. In other words, before applying a mathematical model, unknown model parameters should be selected so that the model performs as well as possible within the constraints imposed by physical conditions or measured data. The selection of the "best" set of model parameters requires some kind of ranking basis. This basis is usually evaluated by a function called "objective function" (Eq. 1). The selection of an objective function and the recommended procedure for calibrating a complicated model are discussed as follows.

4.2 Objective Function

Two different objective functions which are commonly used are given below.

4.2.1 Sum of Squares of Deviations

This objective function is defined by the following equation

$$F = \sum_{i=1}^{N} [R_i^e \; (X_1, X_2, \ldots X_{N_p}) - R_i^o]^2 \tag{37}$$

in which N is the number of observations, $R_i^e \; (X_1, X_2, \ldots X_{N_p})$ is the estimated system response utilizing the mathematical model and the values of model parameters of $[X_1, X_2, \ldots X_{N_p}]$ for the ith observation, and R_i^o is the measured system response of the ith observation. For example, R_i^e is the estimated water yield at the ith day from a water-balance simulation model and R_i^o is the observed water yield at the ith day.

This objective function is analogous to the residual variance of a regression analysis. Mathematically speaking, equal weights are placed on all of the observations. However, in reality, this tends to place greater weight on the observation with a larger value which can be viewed by the following.

From Eq. 37:

$$\left| \frac{\partial F}{\partial R_i^o} \right| = 2 \left| R_i^o \; (X_1, X_2, \ldots X_{N_p}) - R_i^o \right| \tag{38}$$

in which $||$ is the absolute value.

Equation 38 shows that the effect of the ith observation R_i^o on the value of the objective function F is directly proportional to the absolute difference between the estimated value and the measured value. This value is usually larger for the observation with a larger quantity. Therefore, a greater weight is usually placed on the observation with a larger value. This is often a desirable condition for modeling a hydrologic or hydraulic system because an event with a larger quantity is usually more important in considering a design risk.

4.2.2 Sum of Squares of Logarithmic Deviations

The objective function is given below.

$$F = \sum_{i=1}^{N} [\ln R_i^e \; (X_1, X_2, \ldots, X_{N_p}) - \ln R_i^o]^2 \tag{39}$$

According to Dawdy et al. (1972), the logarithms of observation values are used because the prediction errors are generally more nearly equal in percentage than they are in absolute terms. The logarithmic transformation is meant to make the error of estimation more commensurable for the large and the small observation quantities. This can be explained as follows.

From Eq. 39:

$$\left| \frac{\partial F}{\partial R_i^o} \right| = 2 \left| \frac{R_i^e(X_1, X_2, \ldots X_{N_p})}{R_i^o} - 1 \right| \tag{40}$$

The ratios of the estimated value to the measured value are generally nearly equal. From Eq. 40 it can be shown that the effects of different observations on the value of F are nearly the same. Thus, this objective function makes the error of estimation more commensurable for the large and the small observation quantities. This is desirable when the smaller observation values are as equally important as the larger observation quantities.

4.3 Recommended Procedure

In a complicated mathematical model, there are often too many unknown parameters which need to be calibrated. The larger the number of model parameters the more difficult the calibration problem will be. This is because of more interactions among parameters. It is viable to decompose the optimization problem into various sequential probelms with a smaller number of unknown parameters. This decomposition should be done according to the physical significance and the results of parameter sensitivity. For example, the water and sediment routing model developed by Simons et al. (1975) contains parameters governing various system responses such as water routing and yield, wash load yield, and bed material load routing and yield. The calibration should be made sequentially according to these various system responses. The recommended procedures are given herein.

a) <u>Water Routing and Yield</u>

<u>Step 1</u>. Identify parameters governing the water yield.

Step 2. Based on the results obtained in Step 1, estimate the optimum set of parameters governing the water routing.

Step 3. Let the results obtained in Step 2 be the initial estimate, recalibrate the model considering both the water yield and water routing. The objective function can be assumed as the following

$$F = \theta F_1 + (1-\theta)F_2 \tag{41}$$

in which θ is the weighting factor, F_1 is the objective function representing water yield, and F_2 is the objective function representing water routing. An appropriate value of θ is 0.5.

b) Sediment Routing and Yield

Step 4. Based on the optimum parameters governing water routing and yield, identify parameters governing wash load routing and yield.

Step 5. From the set of parameters obtained in Step 4, estimate the optimum set of parameters governing bed-material load.

Step 6. Let the set of parameters obtained in Step 5 be the initial estimate, find the optimum set of parameters considering both the wash load and the bed-material load routing and yield. A similar objective function to Eq. 41 can be used, i.e.,

$$F = \theta F_3 + (1-\theta) F_4 \tag{42}$$

in which F_3 and F_4 are respectively objective functions representing wash load and bed-material load sediment yield.

An example of the calibration results of the above procedures was given by Simons et al. (1975).

V. SUMMARY

A one-dimensional calibration technique modified from Powell's (1964) unidimensional minimization method is proposed to calibrate one-dimensional models. This unidimensional method is further applied to modify the Rosenbrock's (1960) method for the calibration of models with multiple parameters. This modification shortened computer time compared with the original Rosenbrack's method.

Both one-dimensional and multi-dimensional calibration techniques are formulated to deal with bound constraints (i.e., the upper and lower bounds). These bound constraints are usually imposed on the mathematical models by physical conditions or measured data.

It is found that the objective function based on the sum of squares of deviations generally places more weight on the observations with larger absolute quantities. This would provide a safer design considering a risk analysis. The objective function based on the sum of squares of logarithmic deviations would make the error of estimation more commensurate for the large and the small observation quantities. This would be desirable when the smaller observation values are as equally important as the larger observation quantities.

For calibrating a complicated system, it is recommended that the calibration problem be decomposed into various sequential calibration problems with a much smaller number of unknown parameters.

REFERENCES

1. Beard, L.R., 1967, "Optimization Techniques for Hydrologic Engineering," Water Resources Research, Vol. 3, No. 3, pp. 807-815.

2. Dawdy, D.R. and O"Donnell, T., 1965, "Mathematical Models of Catchment Behavior," Journal of the Hydraulics Division, ASCE, Vol. 91, No. HY4, Proceedings Paper 4410, July, pp. 123-137.

3. Dawdy, D.R., Litchy, R.W. and Bergmann, J. M., 1972, "A Rainfall-Runoff Simulation Model for Estimation of Flood Peaks for Small Drainage Basin," USGS Professional Paper 506-B.

4. Himmelblau, D.M., 1972, Applied Nonlinear Programming, McGraw-Hill, New York.

5. Ibbitt, R.P. and O'Donnell, T., 1971, "Fitting Methods for Conceptual Catchment Models," Journal of the Hydraulics Division, ASCE, Vol. 17, No. HY9, September, pp. 1331-1342.

6. Labadie, J.W. and Dracup, J.A., 1969, "Optimal Identification of Lumped Watershed Model," Water Resources Research, Vol. 5, No. 3, pp. 583-590.

7. Liou, E.Y., 1970, "OPSET: Program for Computerized Selection of Watershed Paprmeter Values for the Stanford Watershed Model," Water Resources Research Institute, Report 34, University of Kentucky, Lexington.

8. Overton, D.E., 1968, "A Least-Squares Hydrograph Analysis of Complex Storms on Small Agricultural Watersheds," Water Resources Research, Vol. 4, No. 5, pp. 955-963.

9. Palmer, J.R., 1969, "An Improved Procedure for Orthogonalizing the Search Vectors in Rosenbrock's and Swann's Direct Search Optimization Methods," The Computer Journal, Vol. 12, pp. 69-71.

10. Powell, M.J.D., 1964, "An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives," The Computer Journal, Vol. 7, pp. 155-162.

11. Rosenbrock, H.H., 1960, "An Automatic Method for Finding the Greatest or Least Value of a Function," The Computer Journal, Vol. 3, pp. 175-184.

12. Simons, D.B., Li, R.M., and Stevens, M.A., 1975, "Development of Models for Predicting Water and Sediment Routing and Yield from Storms on Small Watersheds," Colorado State University Report CER74-75DBS-RML-MAS24. Prepared for USDA Forest Service, Rocky Mountain Forest and Range Experimental Station, Flagstaff, Arizona, August.

13. Tuffuor, S. and Labadie, J.W., 1974, "A Nonlinear Time Variant Rainfall-Runoff Model for Augmenting Monthly Data," Water Resources Research, Vol. 10, No. 6, pp. 1161-1166.

14. Yeh, W.G. and Tauxe, G.W., 1971, "Optimal Identification of Aquifer Diffusivity using Quasi-linearization," Water Resources Research, Vol. 7, No. 4, pp. 955-962.

15. Zangwill, W.I., 1967, "Minimizing a Function without Calculating Derivatives," The Computer Journal, Vol. 10, pp. 293-296.

APPENDIX A

LISTING OF COMPUTER PROGRAM UNIMO

PROGRAM UNIMO (INPUT,OUTPUT)

```
      PROGRAM UNIMO (INPUT,OUTPUT)
C
C     THIS PROGRAM SOLVES ONE-DIMENSIONAL CONSTRAINED MINIMIZATION
C     PROBLEM BY SUCCESSIVE QUADRATIC APPROXIMATION
C     THE CONSTRAINTS ARE THE UPPER AND LOWER BOUNDS OF THE VECTOR
C     THE USER MUST SUPPLY A SUBROUTINE OBJECT FOR EVALUATION OF THE
C     OBJECTIVE FUNCTION
C     NOTATIONS FOR INPUT AND OUTPUT INFORMATION
C     TITLE = ALPHABETICAL OR NUMERICAL IDENTIFICATION OF THE PROBLEM
C     MST = MAXIMUM LIMIT OF NUMBER OF STAGE SEARCH
C     IPT = NUMERICAL IDENTIFICATION FOR OUTPUT CONTROL
C     IPT = 0 --- ONLY THE FINAL ANSWER IS PRINTED
C     IPT = 1 --- INTERMEDIATE VALUES OF EACH STAGE SEARCH ARE PRINTED
C     XA = INITIAL GUESS OF THE VECTOR
C     DX = INITIAL STEP-SIZE
C     XUPL = UPPER BOUND
C     XLOL = LOWER BOUND
C     EPS = CONVERGENCE TOLERANCE
C
      DIMENSION E(3), Y(3), TITLE(20)
C
C     INPUT AND OUTPUT NECESSARY INFORMATION
C
      READ 118, TITLE
      PRINT 119, TITLE
      READ 120, MST,IPT,XA,DX,XUPL,XLOL,EPS
      PRINT 121, XA,XUPL,XLOL,EPS
C
C     STARTING OF STAGE SEARCH
C
      NEF=0
      NS=0
      CALL OBJECT (VALUE,NEF,XA)
      A=VALUE
      XB=XA+DX
      CALL OBJECT (VALUE,NEF,XB)
      B=VALUE
C
C     DETERMINE THE THIRD POINT REQUIRED FOR APPROXIMATION
C
      IF (A.GT.B) GO TO 104
  101 XC=XA-DX
      IF (XC.GE.XLOL) GO TO 102
      XC=XLOL
  102 CALL OBJECT (VALUE,NEF,XC)
      C=VALUE
      Y(1)=XC
      Y(2)=XA
      Y(3)=XB
      E(1)=C
      E(2)=A
      E(3)=B
      IF (C.LT.A) GO TO 103
      XINF=XA
      FINF=A
```

PROGRAM UNIMO (INPUT,OUTPUT)

```
      GO TO 107
  103 XINF=XC
      FINF=C
      GO TO 107
  104 XC=XA+2.*DX
      IF (XC.LE.XUPL) GO TO 105
      XC=XUPL
  105 CALL OBJECT (VALUE,NEF,XC)
      C=VALUE
      Y(1)=XA
      Y(2)=XB
      Y(3)=XC
      E(1)=A
      E(2)=B
      E(3)=C
      IF (C.LT.B) GO TO 106
      XINF=XB
      FINF=B
      GO TO 107
  106 XINF=XC
      FINF=C
C
C     ELIMINATE PREMATURE TERMINATION DUE TO EQUAL VALUES AT TWO END
C     POINTS IN THE FIRST SEARCH
C
  107 DEF=E(1)-E(3)
      IF (NS.GT.0.OR.ABS(DEF).GT.EPS) GO TO 108
      DX=0.5*DX
      Y(2)=Y(1)+DX
      CALL OBJECT (VALUE,NEF,Y(2))
      E(2)=VALUE
      Y(3)=XINF
      E(3)=FINF
      DEF=E(1)-E(3)
      IF (E(2).GT.FINF) GO TO 108
      XINF=Y(2)
      FINF=E(2)
C
C     CHECK THE CONVEXITY OF THE QUADRATIC FUNCTION
C
  108 A1=(Y(1)-Y(2))*(Y(2)-Y(3))*(Y(1)-Y(3))
      IF (ABS(A1).EQ.0.) GO TO 109
      A2=E(1)*(Y(2)-Y(3))+E(2)*(Y(3)-Y(1))+E(3)*(Y(1)-Y(2))
      SA=A2/A1
      IF (SA.GE.0.) GO TO 110
      DX=Y(3)-Y(1)
      XA=Y(1)
      A=E(1)
      XP=Y(3)
      B=E(3)
      IF (DEF.GT.0.) GO TO 104
      GO TO 101
  109 XSTA=XINF
      FSTA=FINF
      GO TO 117
```

PROGRAM UNIMO (INPUT,OUTPUT)

```
C
C     DETERMINE THE MINIMUM OF THE QUADRATIC FUNCTION
C
  110 SB=(E(1)-E(2))/(Y(1)-Y(2))-SA*(Y(1)+Y(2))
      XSTA=-SB/(2.*SA)
      IF (XSTA.GE.XLOL.AND.XSTA.LE.XUPL) GO TO 112
      IF (DEF.GT.0.) GO TO 111
      XSTA=XLOL
      GO TO 112
  111 XSTA=XUPL
  112 NS=NS+1
      CALL OBJECT (VALUE,NEF,XSTA)
      FSTA=VALUE
      IF (FSTA.LE.FINF) GO TO 113
      XTEM=XSTA
      XSTA=XINF
      XINF=XTEM
      FTEM=FSTA
      FSTA=FINF
      FINF=FTEM
  113 IF (IPT.EQ.0) GO TO 114
      PRINT 122
      PRINT 123, NS
      PRINT 122
      PRINT 124, XSTA,FSTA
C
C     CHECK IF THE VALUE IS SATISFIED WITH CONVERGENCE TOLERANCE
C
  114 DX=ABS(XINF-XSTA)
      IF (DX.EQ.0.) GO TO 117
      IF ((1.-FSTA/FINF).LE.EPS) GO TO 117
      IF (NS.LT.MST) GO TO 115
      PRINT 122
      PRINT 125, MST
      PRINT 124, XSTA,FSTA
      STOP
  115 IF (XSTA.GT.XINF) GO TO 116
      XA=XSTA
      A=FSTA
      XB=XINF
      B=FINF
      GO TO 101
  116 XA=XINF
      A=FINF
      XB=XSTA
      B=FSTA
      GO TO 104
C
C     A MINIMUM HAS BEEN FOUND
C
  117 PRINT 122
      PRINT 126, NS,NEF
      PRINT 127, FSTA,XSTA
      STOP
C
```

PROGRAM UNIMO (INPUT,OUTPUT)

```
  118 FORMAT (20A4)                                                       UNI 1660
  119 FORMAT (1H1/////40X,20A4)                                           UNI 1670
  120 FORMAT (2I10,4F10.5,E10.3)                                          UNI 1680
  121 FORMAT (//35X,39HTHE INITIAL VECTOR CHOSEN BY THE USER =,F10.5//41UNI 1690
     1X,27HUPPER LIMIT OF THE VECTOR =,F10.5//41X,27HLOWER LIMIT OF THE UNI 1700
     2VECTOR =,F10.5//44X,23HCONVERGENCE TOLERANCE =,E10.3)             UNI 1710
  122 FORMAT (/40X,40H****************************************)        UNI 1720
  123 FORMAT (//48X,18HSTAGE SEARCH -----,I5)                             UNI 1730
  124 FORMAT (//45X,20HTHE CURRENT VECTOR =,F10.5//34X,32HTHE CURRENT OBUNI 1740
     1JECTIVE FUNCTION =,E20.8)                                           UNI 1750
  125 FORMAT (//40X,18HDO NOT CONVERGE IN,I5,5X,14HSTAGE SEARCHES)        UNI 1760
  126 FORMAT (//48X,24HA MINIMUM HAS BEEN FOUND//41X,30HTOTAL NUMBER OF UNI 1770
     1STAGE SEARCH =,I5//39X,37HTOTAL NUMBER OF FUNCTION EVALUATION =,I5UNI 1780
     2)                                                                   UNI 1790
  127 FORMAT (//38X,23HOPTIMIZATION FUNCTION =,E20.8//48X,14HFINAL VECTOUNI 1800
     1R =,F10.5)                                                          UNI 1810
C                                                                         UNI 1820
      END                                                                 UNI 1830
```

SUBROUTINE OBJECT (VALUE,NEF,X)

```
      SUBROUTINE OBJECT (VALUE,NEF,X)
C
C     THIS FUNCTION EVALUATES THE VALUE OF THE OBJECTIVE FUNCTION
C
      NEF=NEF+1
      VALUE=(1.-X)**2+(1.-X*X)**2
      RETURN
C
      END
```

APPENDIX B

LISTING OF COMPUTER PROGRAM BROSEN

PROGRAM BROSEN (INPUT,OUTPUT)

```
      PROGRAM BROSEN (INPUT,OUTPUT)
C
C     THIS PROGRAM SOLVES CONSTRAINED MINIMIZATION PROBLEM
C     THE CONSTRAINTS ARE LIMITED TO BOUND CONSTRAINTS, OR UPPER AND
C     LOWER BOUND
C     THE SOLUTION TECHNIQUE IS A MIX APPLICATION OF THE ORIGINAL
C     ROSENBROCK METHOD, POWELL MINIMIZATION, AND PALMER VERSION OF
C     GENERATING NEW SEARCH DIRECTIONS
C     THE USER MUST SUPPLY A SUBROUTINE OBJECT FOR EVALUATION OF THE
C     OBJECTIVE FUNCTION
C     NOTATIONS FOR INPUT AND OUTPUT INFORMATION
C     TITLE = ALPHABETICAL OR NUMERICAL IDENTIFICATION OF THE PROBLEM
C     N = NUMBER OF VARIABLES
C     MST = MAXIMUM LIMIT OF NUMBER OF STAGE SEARCH
C     MCL = MAXIMUM LIMIT OF NUMBER OF CYCLE SEARCH
C     IPT = NUMERICAL IDENTIFICATION FOR OUTPUT CONTROL
C     IPT = 0 --- ONLY THE FINAL ANSWER IS PRINTED
C     IPT = 1 --- INTERMEDIATE VALUES OF EACH STAGE SEARCH ARE PRINTED
C     IPT = 2 --- INTERMEDIATE VALUES OF EACH CYCLE SEARCH ARE PRINTED
C     EPS = CONVERGENCE TOLERANCE BASED ON THE CHANGE OF OBJECTIVE
C           FUNCTION
C     EPX = CONVERGENCE TOLERANCE FOR CYCLE SEARCH
C     V = INITIAL GUESS OF THE VECTOR
C     VUP = UPPER LIMIT OF THE VECTOR
C     VLO = LOWER LIMIT OF THE VECTOR
C     X = NORMALIZED INITIAL GUESS OF THE VECTOR
C     PO = OPTIMUM VALUE OF THE OBJECTIVE FUNCTION
C     NEF = NUMBER OF FUNCTION EVALUATION
C     NS = NUMBER OF STAGE SEARCH
C
      DIMENSION A(10), B(10), C(10), D(10), Z(10), TITLE(20)
      COMMON DL,DX,PO,VALUE,N,NEF,S(10,10),X(10),V(10),VUP(10),VLO(10)
      COMMON /UNI/ MCL,EPX
C
C     INPUT AND OUTPUT NECESSARY INFORMATION
C
      READ 120, TITLE
      PRINT 121, TITLE
      READ 122, N,MST,MCL,IPT,EPS
      PRINT 123, N,EPS
      READ 124, (V(I),D(I),VUP(I),VLO(I),I=1,N)
      PRINT 125
      PRINT 126, (I,VUP(I),VLO(I),I=1,N)
      PRINT 127
      PRINT 128, (I,V(I),I=1,N)
      PRINT 119
      PRINT 128, (I,D(I),I=1,N)
C
      EPX=10.*EPS
C
C     NORMALIZE THE VECTOR
C
      DO 101 I=1,N
         X(I)=(V(I)-VLO(I))/(VUP(I)-VLO(I))
  101 CONTINUE
```

PROGRAM BROSEN (INPUT,OUTPUT)

```
C
C     SET THE INITIAL SEARCH DIRECTION
C
      DO 103 I=1,N
         DO 102 J=1,N
            S(I,J)=0.
            IF (J.EQ.I) S(I,J)=1.
  102    CONTINUE
  103 CONTINUE
C
C     STARTING OF STAGE SEARCH
C
      NS=0
      NEF=0
      CALL OBJECT (1,0.)
      PO=VALUE
  104 NS=NS+1
      OBJ=PO
      IF (IPT.EQ.0) GO TO 105
      PRINT 129
      PRINT 130, NS
  105 DO 107 I=1,N
         DX=D(I)
         CALL UNIMO (I)
         IF (IPT.NE.2) GO TO 106
         PRINT 131, I
         PRINT 132, PO
         PRINT 128, (J,V(J),J=1,N)
  106    Z(I)=DL
         D(I)=ABS(DL)
  107 CONTINUE
C
C     CHECK IF THE RESULT IS SATISFIED WITH THE PREASSIGNED CONVERGENCE
C     TOLERANCE
C
      IF ((1.-PO/OBJ).LE.EPS) GO TO 118
C
C     CHECK IF THE NUMBER OF STAGE SEARCH GREATER THAN ASSIGNED LIMIT
C
      IF (NS.LT.MST) GO TO 108
      PRINT 129
      PRINT 133, MST
      PRINT 132, PO
      PRINT 128, (I,V(I),I=1,N)
      STOP
  108 PRINT 129
      PRINT 134, NEF
      PRINT 132, PO
      PRINT 128, (I,V(I),I=1,N)
C
C     CALCULATE NEW SEARCH DIRECTION FOR NEXT STAGE SEARCH
C     PALMERS VERSION IS USED TO COMPUTE THE NEW DIRECTION
C
      DO 117 I=1,N
         SUMA=0.
```

PROGRAM BROSEN (INPUT,OUTPUT)

```
      DO 110 J=1,N
         A(J)=0.
         DO 109 K=I,N
            A(J)=A(J)+Z(K)*S(K,J)
109      CONTINUE
         SUMA=SUMA+A(J)**2
110   CONTINUE
      AA=SQRT(SUMA)
      IF (AA.EQ.0.) GO TO 104
      IF (I.EQ.1) GO TO 112
      IF (ABS(Z(I-1)).LE.EPS) GO TO 114
      DA=1./SQRT(AB**2-AA**2)
      RA=AB/AA
      CA=DA*RA
      CB=DA/RA
      DO 111 J=1,N
         C(J)=S(I,J)
         S(I,J)=A(J)*CA-B(J)*CB
         B(J)=A(J)
111   CONTINUE
      GO TO 116
112   DO 113 J=1,N
         C(J)=S(I,J)
         S(I,J)=A(J)/AA
         B(J)=A(J)
113   CONTINUE
      GO TO 116
114   DO 115 J=1,N
         CTEM=S(I,J)
         S(I,J)=C(J)
         C(J)=CTEM
         B(J)=A(J)
115   CONTINUE
116   AB=AA
117 CONTINUE
    GO TO 104
C
C   A MINIMUM HAS BEEN FOUND
C
118 PRINT 129
    PRINT 135, NS,NEF
    PRINT 136, PO
    PRINT 128, (I,V(I),I=1,N)
    STOP
C
119 FORMAT (//47X,25HTHE CHOSEN STEP SIZES ARE)
120 FORMAT (20A4)
121 FORMAT (1H1/////40X,20A4)
122 FORMAT (4I10,E10.3)
123 FORMAT (//47X,21HNUMBER OF VARIABLES =,I5//44X,23HCONVERGENCE TOLE
   1RANCE =,E10.3)
124 FORMAT (4F10.5)
125 FORMAT (//44X,33HUPPER AND LOWER BOUNDS OF VECTORS)
126 FORMAT (/10X,4(I6,2F12.5))
127 FORMAT (//40X,40HTHE INITIAL VECTOR CHOSEN BY THE USER IS)
```

PROGRAM BROSEN (INPUT,OUTPUT)

```
  128 FORMAT (/18X,5(I5,F12.5))                                           BRO 1660
  129 FORMAT (/40X,40H****************************************)           BRO 1670
  130 FORMAT (//48X,18HSTAGE SEARCH -----,I5)                             BRO 1680
  131 FORMAT (//40X,34HCYCLE SEARCH ALONG DIRECTION -----,I5)             BRO 1690
  132 FORMAT (//34X,32HTHE CURRENT OBJECTIVE FUNCTION =,E20.8//50X,21HTHBRO 1700
     1E CURRENT VECTOR IS)                                                BRO 1710
  133 FORMAT (//40X,18HDO NOT CONVERGE IN,I5,5X,14HSTAGE SEARCHES)        BRO 1720
  134 FORMAT (//36X,43HTHE CURRENT NUMBER OF FUNCTION EVALUATION =,I5)    BRO 1730
  135 FORMAT (//48X,24HA MINIMUM HAS BEEN FOUND//41X,30HTOTAL NUMBER OF BRO 1740
     1STAGE SEARCH =,I5//39X,37HTOTAL NUMBER OF FUNCTION EVALUATION =,I5BRO 1750
     2)                                                                   BRO 1760
  136 FORMAT (//38X,23HOPTIMIZATION FUNCTION =,E20.8//50X,15HFINAL VECTOBRO 1770
     1R IS)                                                               BRO 1780
C                                                                         BRO 1790
      END                                                                 BRO 1800
```

SUBROUTINE UNIMO (IP)

```
      SUBROUTINE UNIMO (IP)
C
C     THIS SUBROUTINE DETERMINES THE OPTIMAL STEP SIZE ALONG A DIRECTION
C
      DIMENSION E(3), Y(3)
      COMMON DL,DX,PO,VALUE,N,NEF,S(10,10),X(10),V(10),VUP(10),VLO(10)
      COMMON /UNI/ MCL,EPX
C
C     SET UP UPPER AND LOWER LIMITS
C
      XUPL=1.0E+10
      XLOL=-1.0E+10
      DO 102 I=1,N
         IF (S(IP,I).EQ.0.) GO TO 102
         IF (S(IP,I).LT.0.) GO TO 101
         XTEM=(VUP(I)-V(I))/S(IP,I)
         IF (XTEM.LT.XUPL) XUPL=XTEM
         XTEM=(VLO(I)-V(I))/S(IP,I)
         IF (XTEM.GT.XLOL) XLOL=XTEM
         GO TO 102
  101    XTEM=(VUP(I)-V(I))/S(IP,I)
         IF (XTEM.GT.XLOL) XLOL=XTEM
         XTEM=(VLO(I)-V(I))/S(IP,I)
         IF (XTEM.LT.XUPL) XUPL=XTEM
  102 CONTINUE
      NC=0
      XA=0.
      A=PO
      XB=XA+DX
      IF (XB.LE.XUPL) GO TO 103
      XB=XUPL
      DX=XB
  103 CALL OBJECT (IP,XB)
      B=VALUE
C
C     DETERMINE THE THIRD POINT REQUIRED FOR APPROXIMATION
C
      IF (A.GT.B) GO TO 107
  104 XC=XA-DX
      IF (XC.GE.XLOL) GO TO 105
      XC=XLOL
  105 CALL OBJECT (IP,XC)
      C=VALUE
      Y(1)=XC
      Y(2)=XA
      Y(3)=XB
      E(1)=C
      E(2)=A
      E(3)=B
      IF (C.LT.A) GO TO 106
      XINF=XA
      FINF=A
      GO TO 110
  106 XINF=XC
      FINF=C
```

SUBROUTINE UNIMO (IP)

```
      GO TO 110
  107 XC=XA+2.*DX
      IF (XC.LE.XUPL) GO TO 108
      XC=XUPL
  108 CALL OBJECT (IP,XC)
      C=VALUE
      Y(1)=XA
      Y(2)=XB
      Y(3)=XC
      E(1)=A
      E(2)=B
      E(3)=C
      IF (C.LT.B) GO TO 109
      XINF=XB
      FINF=B
      GO TO 110
  109 XINF=XC
      FINF=C
C
C     ELIMINATE PREMATURE TERMINATION DUE TO EQUAL VALUES AT TWO END
C     POINTS IN THE FIRST SEARCH
C
  110 DEF=E(1)-E(3)
      IF (NC.GT.0.OR.ABS(DEF).GT.EPX) GO TO 111
      DX=0.5*DX
      Y(2)=Y(1)+DX
      CALL OBJECT (IP,Y(2))
      E(2)=VALUE
      Y(3)=XINF
      E(3)=FINF
      DEF=E(1)-E(3)
      IF (E(2).GT.FINF) GO TO 111
      XINF=Y(2)
      FINF=E(2)
C
C     CHECK THE CONVEXITY OF THE QUADRATIC FUNCTION
C
  111 A1=(Y(1)-Y(2))*(Y(2)-Y(3))*(Y(1)-Y(3))
      IF (ABS(A1).EQ.0.) GO TO 112
      A2=E(1)*(Y(2)-Y(3))+E(2)*(Y(3)-Y(1))+E(3)*(Y(1)-Y(2))
      SA=A2/A1
      IF (SA.GE.0.) GO TO 113
      DX=Y(3)-Y(1)
      XA=Y(1)
      A=E(1)
      XB=Y(3)
      B=E(3)
      IF (DEF.GT.0.) GO TO 107
      GO TO 104
  112 XSTA=XINF
      FSTA=FINF
      GO TO 119
C
C     DETERMINE THE MINIMUM OF THE QUADRATIC FUNCTION
C
```

SUBROUTINE UNIMO (IP)

```
  113 SB=(E(1)-E(2))/(Y(1)-Y(2))-SA*(Y(1)+Y(2))
      XSTA=-SB/(2.*SA)
      IF (XSTA.GE.XLOL.AND.XSTA.LE.XUPL) GO TO 115
      IF (DEF.GT.0.) GO TO 114
      XSTA=XLOL
      GO TO 115
  114 XSTA=XUPL
  115 NC=NC+1
      CALL OBJECT (IP,XSTA)
      FSTA=VALUE
      IF (FSTA.LE.FINF) GO TO 116
      XTEM=XSTA
      XSTA=XINF
      XINF=XTEM
      FTEM=FSTA
      FSTA=FINF
      FINF=FTEM
  116 IF ((1.-FSTA/FINF).LE.EPX) GO TO 119
      DX=ABS(XINF-XSTA)
      IF (NC.LT.MCL) GO TO 117
      PRINT 121
      PRINT 122, MCL,IP
      STOP
  117 IF (XSTA.GT.XINF) GO TO 118
      XA=XSTA
      A=FSTA
      XB=XINF
      B=FINF
      GO TO 104
  118 XA=XINF
      A=FINF
      XB=XSTA
      B=FSTA
      GO TO 107
C
C     A MINIMUM HAS BEEN FOUND
C
  119 DL=XSTA
      PO=FSTA
      DO 120 I=1,N
         X(I)=X(I)+XSTA*S(IP,I)
         V(I)=VLO(I)+X(I)*(VUP(I)-VLO(I))
  120 CONTINUE
      RETURN
C
  121 FORMAT (/40X,40H****************************************)
  122 FORMAT (//28X,18HDO NOT CONVERGE IN,I5,5X,36HCYCLE SEARCHES ALONG
     1DIRECTION -----,I5)
C
      END
```

SUBROUTINE OBJECT (IP,Z)

```
      SUBROUTINE OBJECT (IP,Z)
C
C     THIS SUBROUTINE DETERMINES THE VALUE OF OBJECTIVE FUNCTION
C
      DIMENSION T(10), Y(10)
      COMMON DL,DX,PO,VALUE,N,NEF,S(10,10),X(10),V(10),VUP(10),VLO(10)
      NEF=NEF+1
      DO 101 I=1,N
         T(I)=X(I)+Z*S(IP,I)
         Y(I)=VLO(I)+T(I)*(VUP(I)-VLO(I))
  101 CONTINUE
      VALUE=(Y(1)-Y(2))**2+(Y(2)-2.*Y(3))**2+(Y(3)-2.)**2
      RETURN
C
      END
```