NON LINEAR PROGRAMMING

by

R.D. SINGH, Sc. 'E'
NIH, Roorkee

1.0 INTRODUCTION

For real time operation of the reservoir, it is necessary to have real time inflow forecast. For this purpose a lumped conceptual hydrological model is generally needed. When a particular model has been chosen, it has to be fitted and tested on observed data before using it for real time flood forecasting. Even during the forecasting period also it would be desirable to specify the correct choice of parameters so that computed flow give sufficiently close reproduction of the observed flows. In order to estimate the optimum parameters such that the computed flood hydrograph has the close agreement with the observed flood hydrograph an optimisation problem may be formulated. Generally, in simulation of flood hydrograph, sum of the squares of the deviations (S) between observed and computed flood hydrographs is minimised. Here S, which is non-linear function, is known as the objective function.

General mathematical form of an optimisation problem can be formulated as:

Min (or Max) $f(x)$

Subject to

$$h_i(x) = 0, \quad i = 1 \ldots m \qquad \ldots(1)$$
$$g_\gamma(x) \leq 0, \quad j = (m+1) \ldots p$$

Where $f(x)$ denotes the objective function, $h_i(x)$ denote the equality constraints and $g_j(x)$ denote the inequality constraints and $x = [x_1, x_2 \ldots, x_n]$ is a row Vector of the parameters to be optimised. Since $g_j(x) \geq 0$ can be written as $- g_j(x) \leq 0$, inequality constraints can be denoted by $g_j(x) \leq 0$. Here m and p are non-negative integers. If m=o and p=0, then problem (1) is said to be unconstrained problem. If $f(x)$, $h_i(x)$ for i=1...m and $g_j(x)$ for j=(m+1)...p are all linear functions, then problem (1) is said to be a linear programming problem (LPP). On the other hand, an optimisation problem, in which either the objective function and/or even one of the constraints is non linear, is called a non-linear programming problem (NLPP)

Since a non-linear function can have various forms, the development of a general optimisation technique is needed, which may handle all types of non-linear optimisation problem, that arise from real life situations. A large number of computational algorithms is available for solving a general non-linear optimisation problem (Flatcher, 1980, Himmelblau, 1972, Nash, 1984). A particular technique, which is efficient for solving a particular type of problem, may not be so efficient to solve another type of non-linear optimisation problem. So the selection of a particular technique depends on the formulation of the problem and the experience of the analyst. The methods available to solve a non-linear programming problem may be classified in the following two categories:

1. Methods for unconstrained optimisation
2. Methods for constrained optimisation

The problems, in which an objective function f(x) is to be optimised, without satisfying any constraints, are called unconstrained optimisation problems whereas the constrained optimisation problems involve the optimisation of an objective function subject to one or more given constraints. Constrained non-linear programming problems are much harder to solve than unconstrained problems with a comparable number of independent variables and degree of non-linearity, because of the additional requirement that the solution must satisfy the constraints.

When the objective function is not explicitly defined as in the case of most of the hydrological problems, it is necessary to examine the behaviour of the objective function before taking up the solution of the optimisation problem in hand. In general the following aspects are considered:

1.1 Maximization or Minimization Problem

While formulating the optimisation problem, one should ensure whether maximization or minimization of the objective function is to be carried out Fig.1 shows a typical objective function (F) evaluated using the different values for the parameter x. The nature of the objective function (F) indicates that the problem is a one parameter maximization problem. On the other hand, if the nature of the objective function (F) is somewhat similar to Fig.2, it would be a one parameter minimization problem.

<u>FOR MAXIMA (Fig.1)</u>

$$\frac{\partial F}{\partial x} = 0 \qquad \ldots(2)$$

$$\frac{\partial^2 F}{\partial x^2} < 0 \qquad \ldots(3)$$

<u>FOR MINIMA (Fig.2)</u>

$$\frac{\partial F}{\partial x} = 0 \qquad \ldots(4)$$

$$\frac{\partial^2 F}{\partial x^2} > 0 \qquad \ldots(5)$$

If the objective function (F) is evaluated using the parameters x_1 and x_2 and the surface of the objective function (F) contour is such that the optima occurs at hill of the contour (Fig.3), the problem would be two parameters maximization problem. However, in case optima lies in the valley of the contours of the objective function (F) as shown in fig.4 the problem would be two parameters minimization problem. Similarly the objective function can also be visualized on different planes for more than two parameters.

1.2 Parameter Sensitivity

Before taking up the optimisation problem sensitivity analysis is to be performed with an objective to identify the sensitive parameters. The objective function (F_1) shown in fig.5 is less sensitive to the parameter x_1 as the gradients $\frac{\partial F_1}{\partial x_1}$ (rate

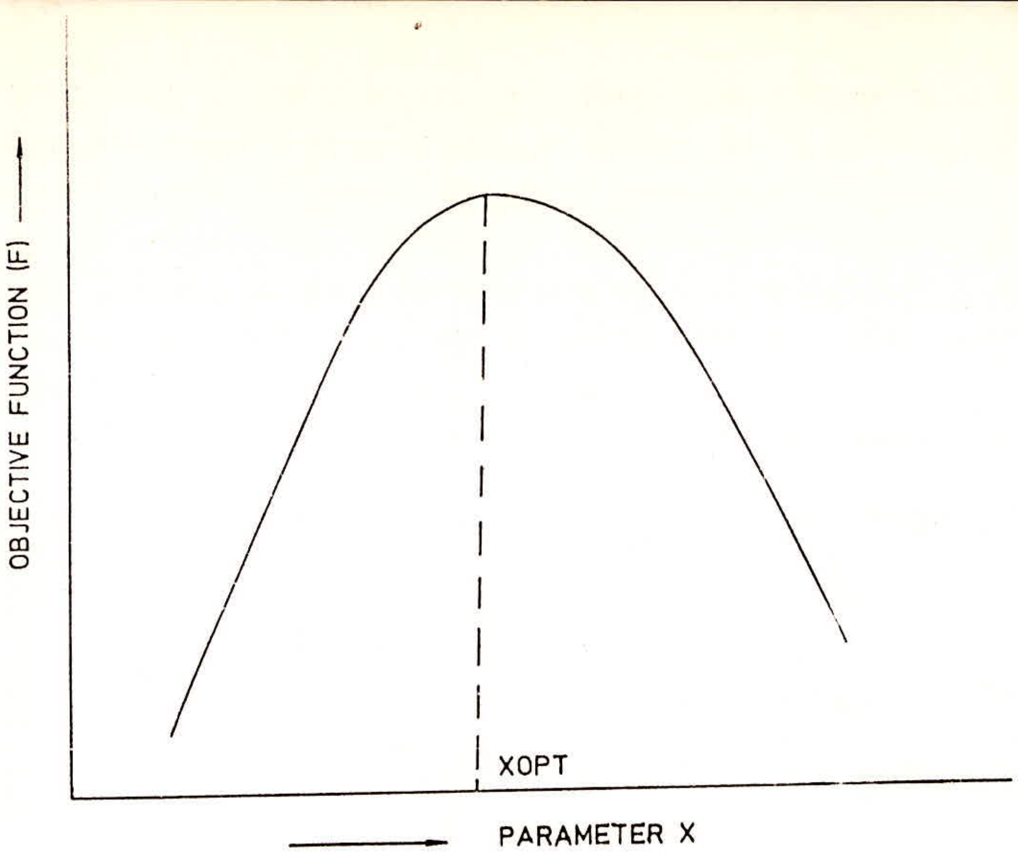

FIG. 1 ONE PARAMETER MAXIMISATION PROBLEM

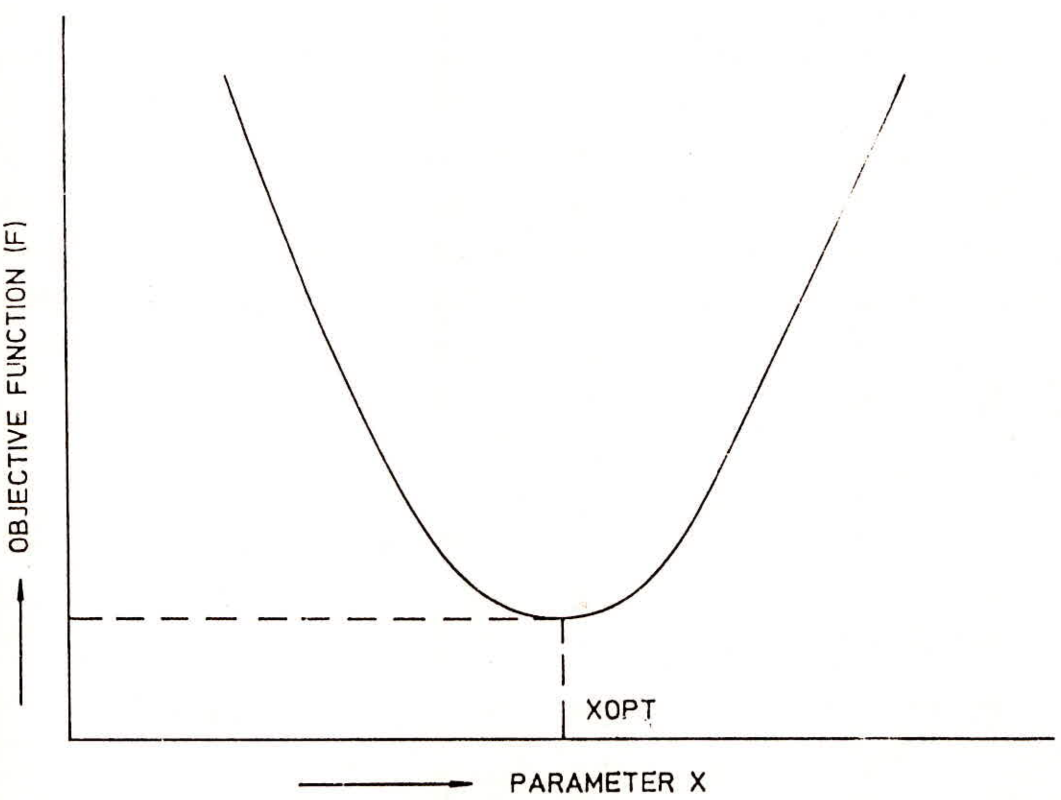

FIG. 2 ONE PARAMETER MINIMISATION PROBLEM

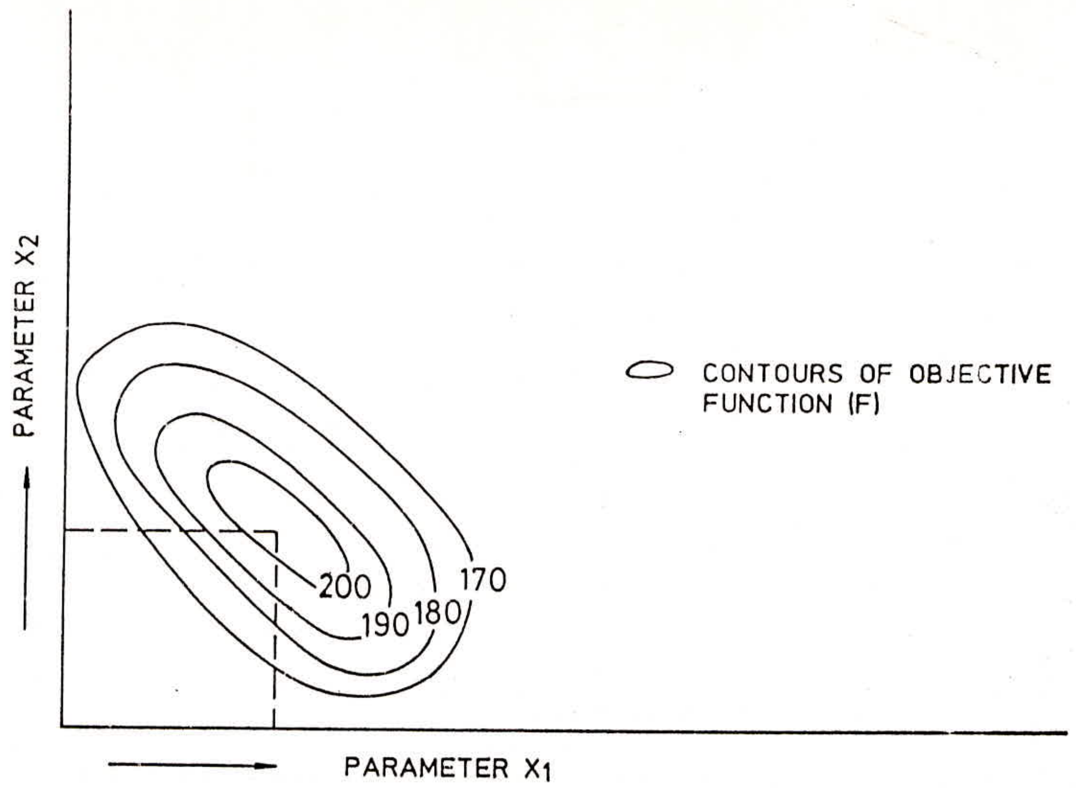

FIG. 3 TWO PARAMETER MAXIMISATION PROBLEM

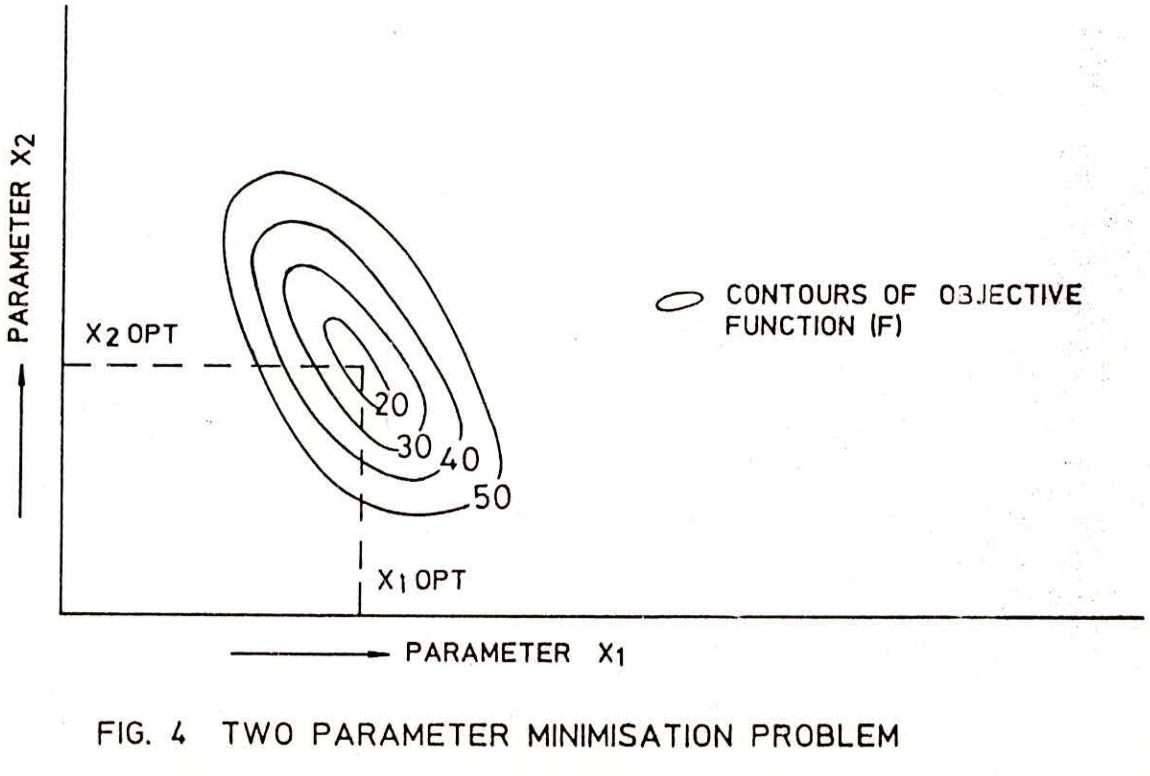

FIG. 4 TWO PARAMETER MINIMISATION PROBLEM

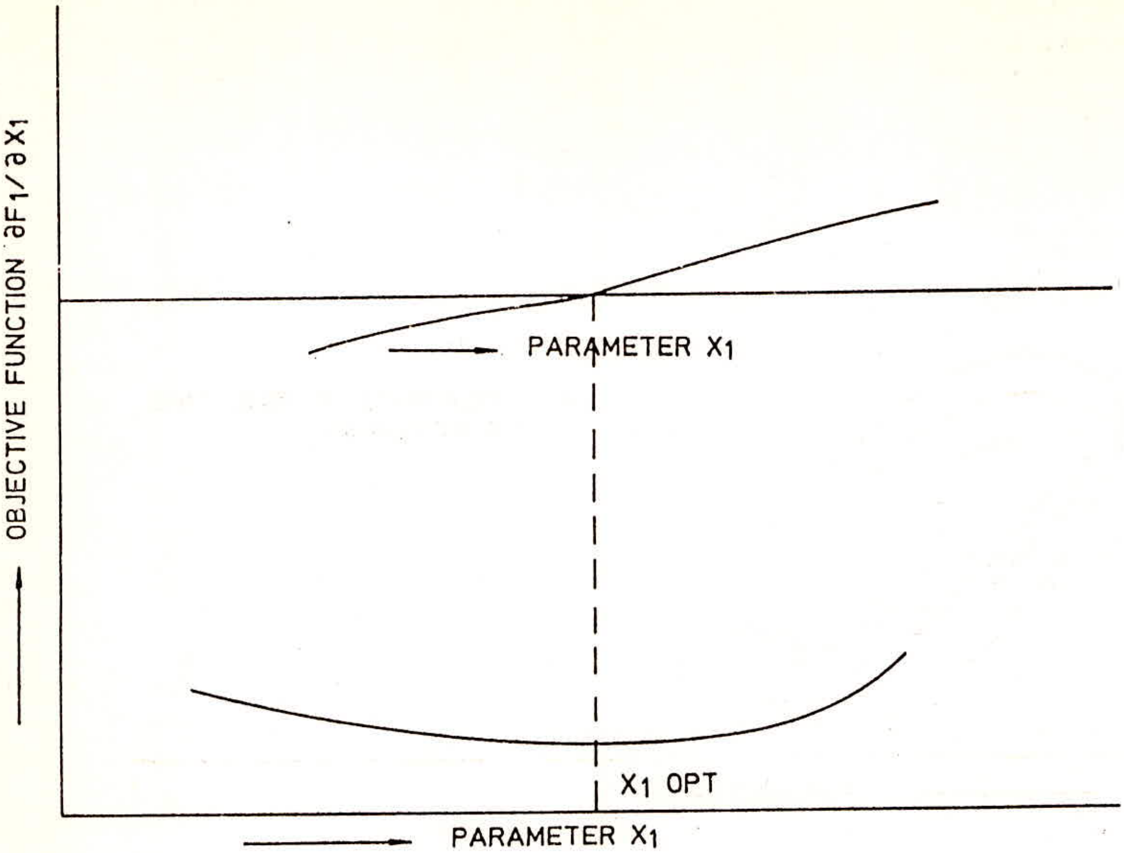

FIG. 5 OBJECTIVE FUNCTION (F) AND SENSITIVITY OF PARAMETER (X_1)

of change of the objective function with respect to the parameter x_1) is less steep than the objective function (F_2) shown in fig.6 which has the higher gradient function $\frac{\partial F_2}{\partial x_2}$. Similar inferences can be drawn from fig.7 for two parameter minimization problem wherein the parameter x_1 is less sensitive in comparison to the parameter x_2. Note that the higher the angle of intersection $(\frac{\partial^2 F}{\partial x^2})$ the better is the definition of the optimum value of the parameter (x). Therefore the value of the second derivative at xOPT is an index of the stability of xOPT. This relationship should be developed so that the index becomes a measure of stability.

During optimisation the sensitive parameters must be dealt cautiously as small change in the parameter values may cause considerable change in the value of the objective function. On the other hand, over estimation or under estimation of least sensitive parameters may not have the pronounced effect over the objective function. Thus to the certain extent one may afford to have the over estimated or under estimated values of the parameters which are least sensitive. On the contrary deviations from the optimum values of the sensitive parameters may give higher value of the objective function.

1.3 Global and Local Optima

Some times the surfaces of objective Function (F) have more than one peak (Fig.8 for one parameter maximization and Fig.10 for two parameter maximization) or more than one trough (Fig.9 for one parameter minimisation problem and Fig.11 for two parameter minimisation problem). In such a situation one may end up at local optima if the incorrect choice of initial parameter values are made for the optimisation runs. The value of the objective function at local optima is more than the global optima in case of minimisation problem and vice versa for the maximisation problem.

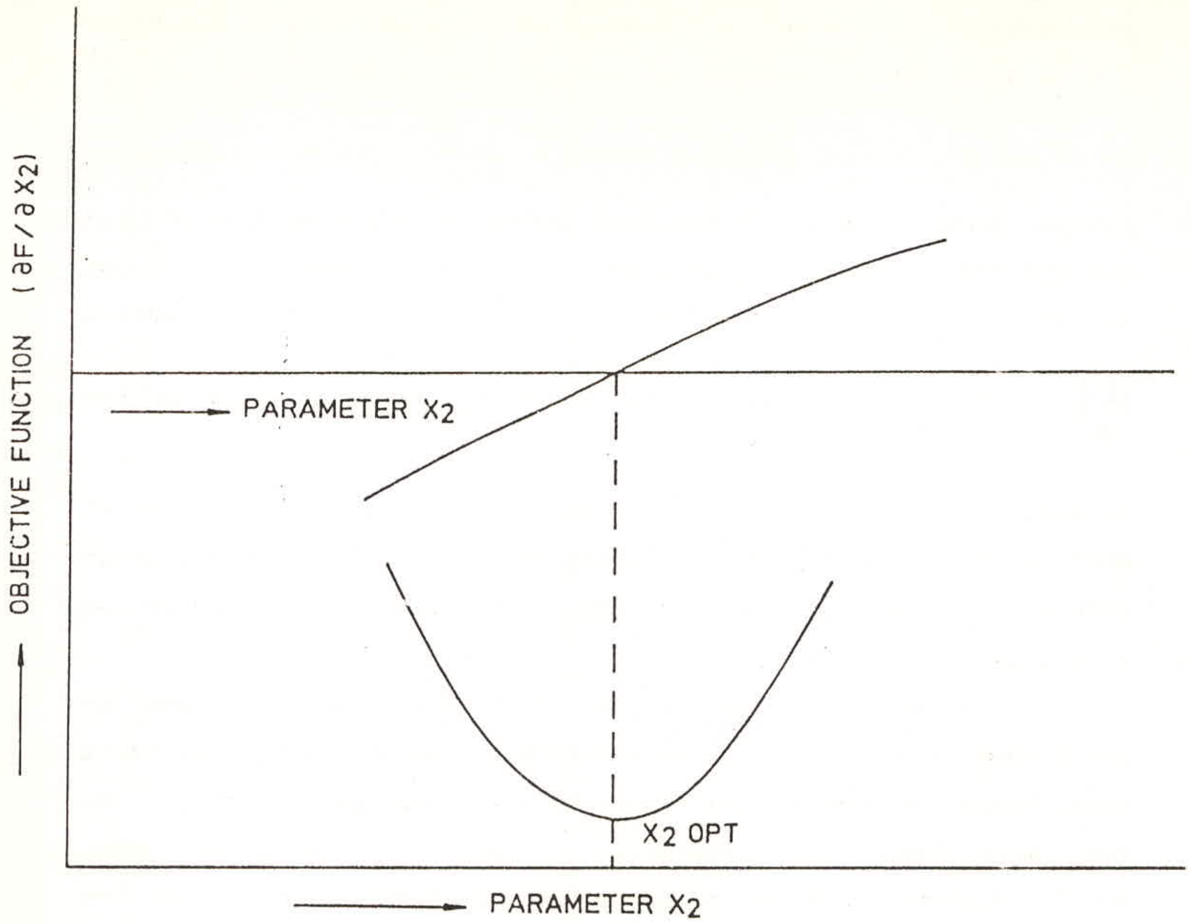

FIG. 6 OBJECTIVE FUNCTION AND SENSITIVITY OF PARAMETER X_2

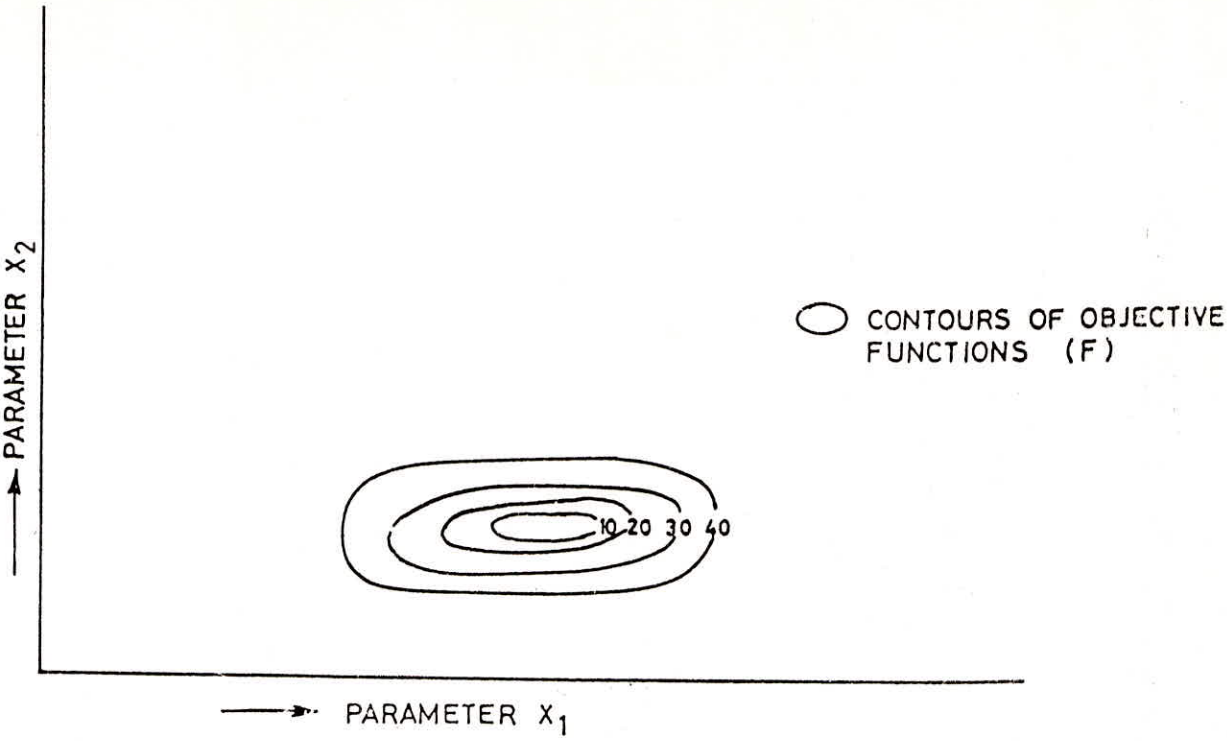

FIG. 7 - TYPICAL OBJECTIVE FUNCTION FOR TWO PARAMETER MINIMISATION PROBLEM

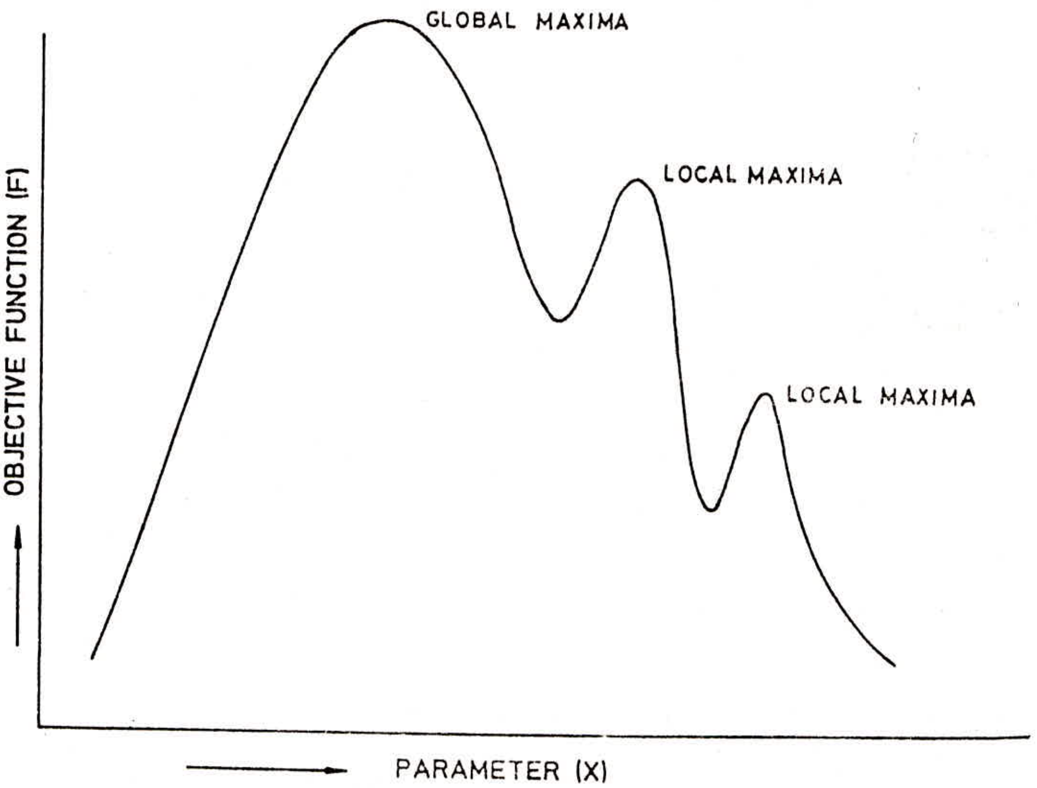

FIG 8 ONE PARAMETER MAXIMISATION PROBLEM (WITH LOCAL MAXIMA)

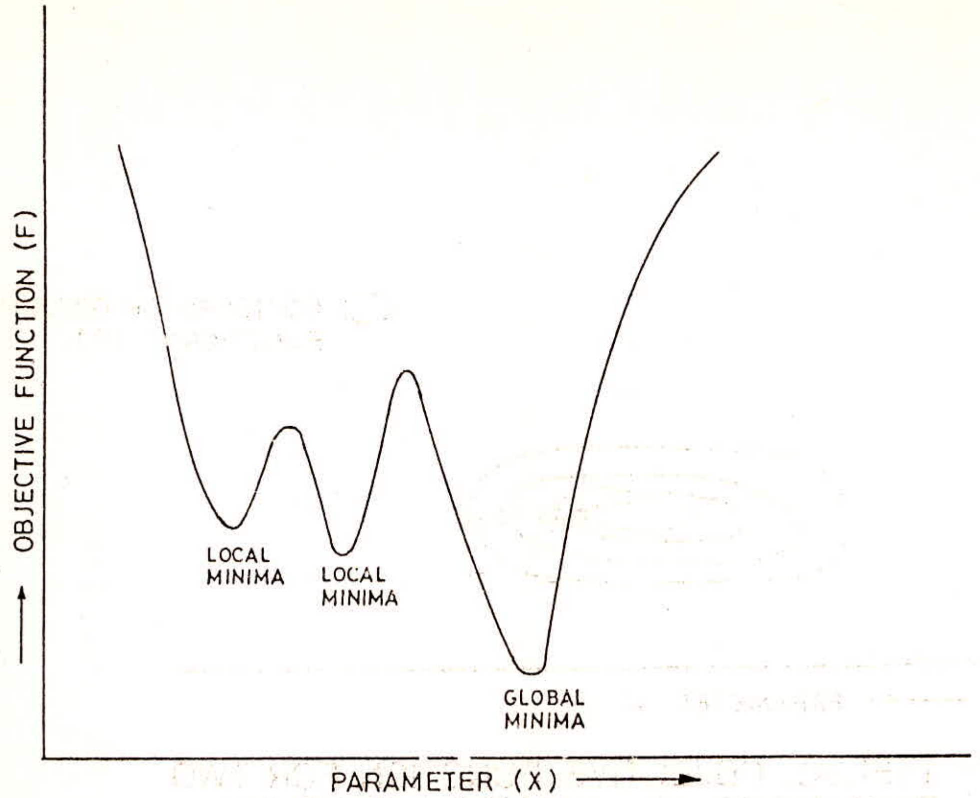

FIG. 9 - ONE PARAMETER MINIMISATION PROBLEM
(WITH LOCAL MINIMA)

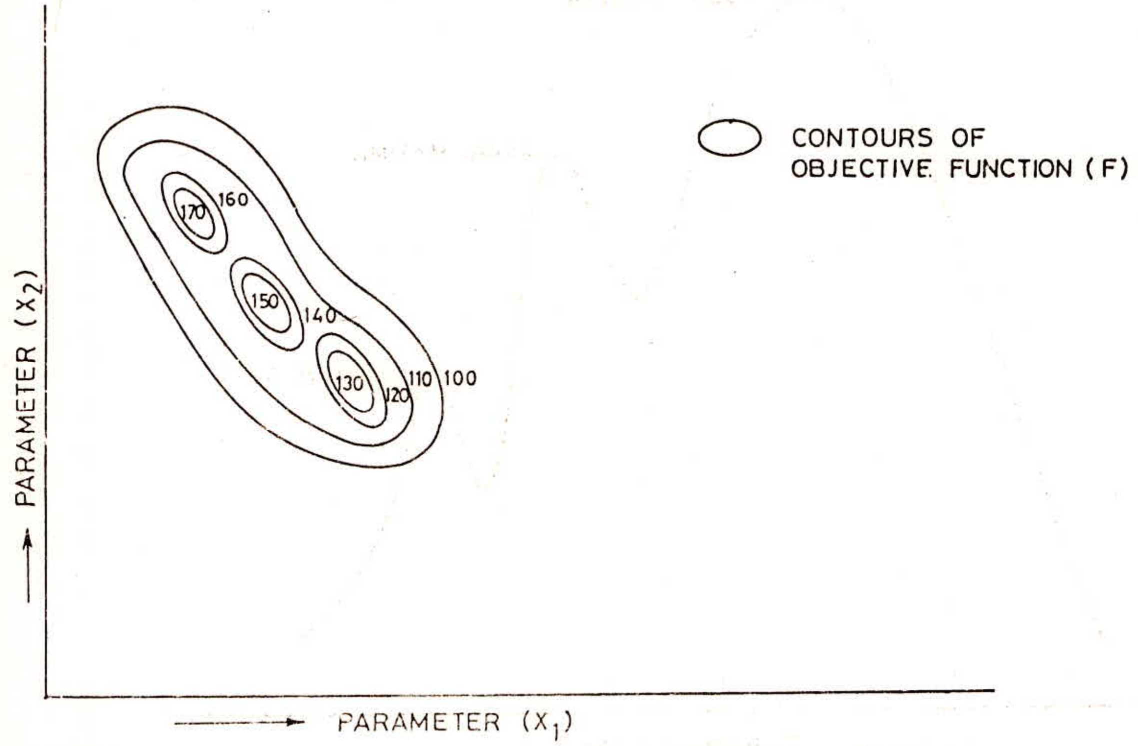

FIG 10 TWO PARAMETER MAX.MISATION PROBLE
(WITH LOCAL MAXIMA)

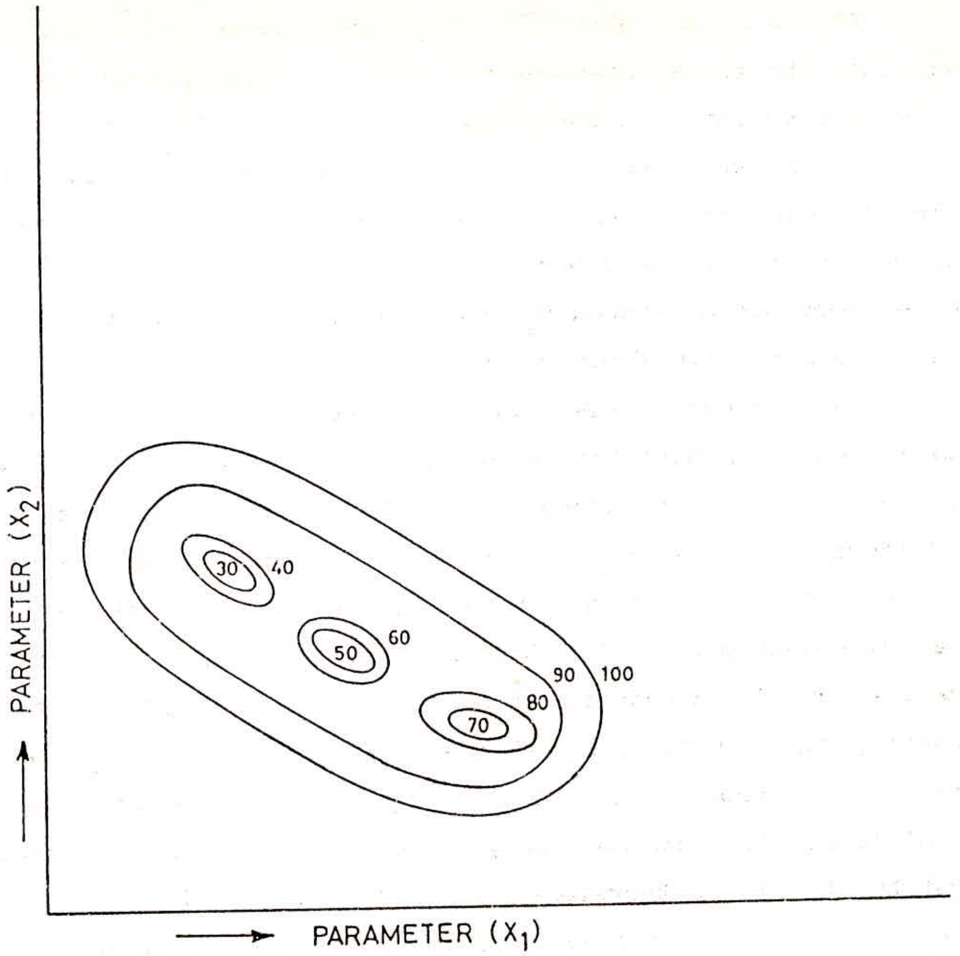

FIG. 11– TWO PARAMETER MINIMISATION PROBLEM
(WITH LOCAL MINIMA)

1.4 Parameter Dependence

As far as possible the parameters involved in optimisation should be independent. If the several parameters are mutually independent then the index of stability of the optimised values may be obtained by considering the second derivative parallel to each axis only. If however substantial dependence exists between two or more parameters this is not sufficient. In fig.4 the dependence between x_1 and x_2 is indicated by the valley in the surface roughly along $x_1 + x_2 = \text{const.}$

The optimum values x1OpT and x2opT may indeed be found and cross sections parallel to the axis may look like those of fig.4. A cross section along the valley, however, would appear very different, indicating that while a function of xOPT and yOPT is well defined, the separate values are not. The occurrence of such a relationship could be discovered only by taking the second derivative in all directions through the minimum point of F. Ordinarily, the introduction of a new part in to a model will enable a better reproduction of the output to be obtained oven if the additional part does not truly represent a physical process in the catchment. It is therefore necessary to have a test to judge whether the observed improvement in fit is accidental or real.

Although various methods are available in literature for solving the non-linear programming problems, here only three methods viz Rosenbrock optimisation Technique, Gauss Newton method and Marquardt Algorithm have been discussed as these methods have been extensively used for solving the non-linear programming problems in hydrology. Rosenbrock optimisation Technique is a multivariable Numerical optimisation technique based on search strategy. Gauss Newton method is generally used to optimise the nonlinear regression equation. Marquardt Algorithm is an improvement over the Gauss Newton method.

2.0 ROSENBROCK OPTIMISATION TECHNIQUE

The procedure is based on the direct search method proposed by H.H. Rosenbrock (1960). No Derivatives are required. The procedure assumes unimodal function, therefore several sets of starting values for the independent variables should be used if it is known that more than one minimum exists or if the shape of the surface is unknown. The algorithm proceeds as follows (minimisation problem):

i. A starting point and initial step sizes, S_i i=1,2....N are picked and the objective function evaluated.

ii. The first variable x_1 is stepped a distance S_1 parallel to the axis, and the function evaluated. If the value of F decreased, the move is termed a success and S_1 increased by a factor α, $\alpha \geq 1.0$. If the value of F increased the move is termed as a failure and S_1 decreased by a factor β, $o < \beta \leq 1.0$, and the direction of movement reversed.

iii. The next variable, x_i, is in turn stepped a distance S_i parallel to the axis. The same acceleration or deceleration and reversal procedure is followed for all variables in consecutive repetitive sequences until a success (decrease in F) and failure (increase in F) have been encountered in all N direction (for minimisation problem).

iv. The axes are then rotated by the following equations. Each rotation of the axes is termed a stage

$$M^{(K+1)}_{i,j} = \frac{D^{(K)}_{i,j}}{\left[\sum_{l=1}^{N} (D^{(K)}_{l,j})^2 \right]^{1/2}} \qquad \ldots(6)$$

where

$$D^{(K)}_{i,1} = A^{(K)}_{i,1} \qquad \ldots(7)$$

$$D^{(K)}_{i,j} = A^{(K)}_{i,j} - \sum_{l=1}^{j-1} \left[\left(\sum_{n=1}^{j} M^{(K+1)}_{n,l} - A^{(K)}_{n,j} \right) - M^{(K+1)}_{i,l} \right] \quad j=2,3\ldots N \qquad \ldots(8)$$

$$A^{K}_{i,j} = \sum_{l=j}^{N} d^{(K)}_{l} - M^{(K)}_{i,l}/ \qquad \ldots(9)$$

where

- i = Variable index
- j = Direction index
- K = Stage index
- d_i = Sum of distances moved in the i direction since last rotation of axes
- $M_{i,j}$ = Direction vector component (normalized)

(v) Search is made in each of the x-directions using the new co-ordinate axes:

$$\text{new } x_i^{(K)} = \text{old } x_i^{(K)} + S_j^{(K)} \quad M_{i,j}^{(K)} \qquad \text{...(10)}$$

(vi) The procedure terminates when the convergence criterion is satisfied. The following convergence criteria may be incorporated in the procedure.

- Maximum number of times program is to evaluate objective function
- Maximum number of times axes are to be rotated
- Number of successive failures encountered in all directions before termination.
- Error in objective function to be reached before the termination of the procedure (taken as difference between current value and previous stage value)

A flow sheet illustrating the above procedure is given in Fig.12.

3.0 GAUSS-NEWTON METHOD

The procedure is based on linearization of the proposed model. A least square objective function is utilized. It solves for the coefficients in a multivariable, non linear regression equation $Y = F(x_1, x_2 ..., x_K; A_1, A_2 ... A_M)$ utilizing N data points for Y_i and $x_{K,i,}$ $i=1,2...N$, $K=1,2,...M$. The method has proved effective where good starting estimates of the unknown coefficients are available. The algorithm proceeds as follows:

i. The model is linearized by expanding Y_i in a Taylor Series about current trial values for the co-efficients and retaining linear terms only.

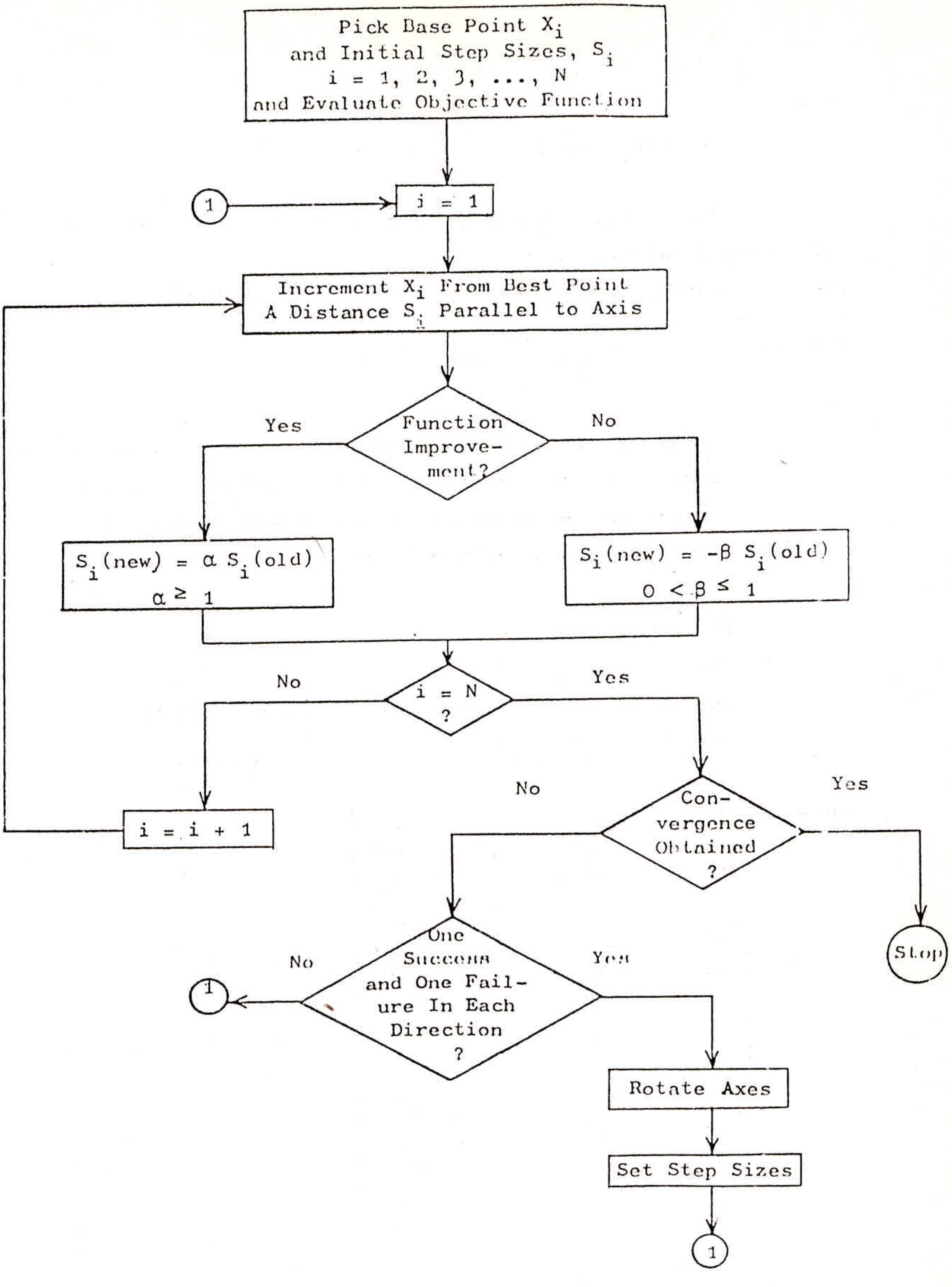

Figure 12. Rosenbrock (ROSENB ALGORITHM) Logic Diagram

$$\hat{Y}_i = \hat{Y}_i^* + \left(\frac{\partial \hat{y}_i}{\partial \hat{A}_1}\right)^* \Delta\hat{A}_1 + \left(\frac{\partial \hat{y}_i}{\partial \hat{A}_2}\right) \Delta\hat{A}_2 + \ldots + \left(\frac{\partial \hat{y}_i}{\partial \hat{A}_M}\right)^* \Delta\hat{A}_M \qquad (11)$$

where

$$\Delta\hat{A}_j = [\hat{A}_j - \hat{A}_j^*], \quad j=1,2\ldots M$$

The asterisk designates quantities evaluated at the initial trial values.

ii. A least square objective function is formulated

Minimize

$$S = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \qquad (12)$$

iii. The linearized model is substituted into the objective function and the "normal equations" formed by setting the partial derivatives of the objective function with respect to each co-efficient equal to zero;

$$\frac{\partial S}{\partial \hat{A}_j} = 0, \quad j = 1,2\ldots M \qquad \ldots(13)$$

The resulting normal equations will be of the form

$$(A^T A)\, \Delta\hat{A} = A^T (y - \hat{y}^*) \qquad \ldots(14)$$

where

$$\begin{bmatrix} \frac{\partial \hat{y}_1}{\partial \hat{A}_1} & \frac{\partial \hat{y}_1}{\partial \hat{A}_2} & \cdots\cdots & \frac{\partial \hat{y}_1}{\partial \hat{A}_M} \\ \frac{\partial \hat{y}_2}{\partial \hat{A}_1} & \frac{\partial \hat{y}_2}{\partial \hat{A}_2} & \cdots\cdots & \frac{\partial \hat{y}_2}{\partial \hat{A}_M} \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \\ \frac{\partial \hat{y}_N}{\partial \hat{A}_1} & \frac{\partial \hat{y}_N}{\partial \hat{A}_2} & \cdots\cdots & \frac{\partial \hat{y}_N}{\partial \hat{A}_M} \end{bmatrix}$$

$$\Delta A = \begin{bmatrix} (\hat{A}_1 - \hat{A}_1^*) \\ (\hat{A}_2 - \hat{A}_2^*) \\ \cdot \\ \cdot \\ (\hat{A}_M - \hat{A}_M^*) \end{bmatrix}, \quad (y - y^*) = \begin{bmatrix} (y_1 - \hat{y}_1^*) \\ (y_2 - \hat{y}_2^*) \\ \cdot \\ \cdot \\ (y_N - \hat{y}_N^*) \end{bmatrix}$$

A^T is the transpose of the A matrix. The derivatives in the A matrix may be evaluated analytically or numerically.

iv. The normal equations are a system of linear algebraic

equations and are solved by an appropriate technique for $\Delta\hat{A}$. The $\Delta\hat{A}$ Vector and S will approach zero as convergence is achieved. If convergence is achieved, the final co-efficients are calculated from

$$\hat{A}_j = \hat{A}_j^* + \Delta\hat{A}_j , \quad j = 1,2......M \qquad ...(15)$$

If convergence is not achieved, $\hat{A}*$ is updated by replacing the old values by the new values and the process repeated.

A flow chart illustrating the above procedure is given in Fig.13.

4.0 MARQUARDT ALGORITHM

The procedure was proposed by Marquardt (1963) as an extension of the Gauss-Newton method to allow for convergence with relatively poor starting guesses for the unknown co-efficients. A least square objective function is utilized. In this method, the Gauss-Newton normal equations are modified by adding a factor λ

$$[A^T A + \lambda I] \; \Delta\hat{A} = A^T \; (Y - \hat{Y}^*) \qquad ...(16)$$

Where I is the identity matrix. Thus λ is added to each term of the main diagonal of the $A^T A$ matrix. It can be shown that when approaches +∞, Marquardt's method is identical to steepest Descent (Jacoby et.al., 1972). When λ equal zero, the technique reduces to Gauss Newton. In general a Steepest Descent Procedure would be expected to converge for poor starting values but requires a lengthy solution time. Gauss Newton, on the other hand, will

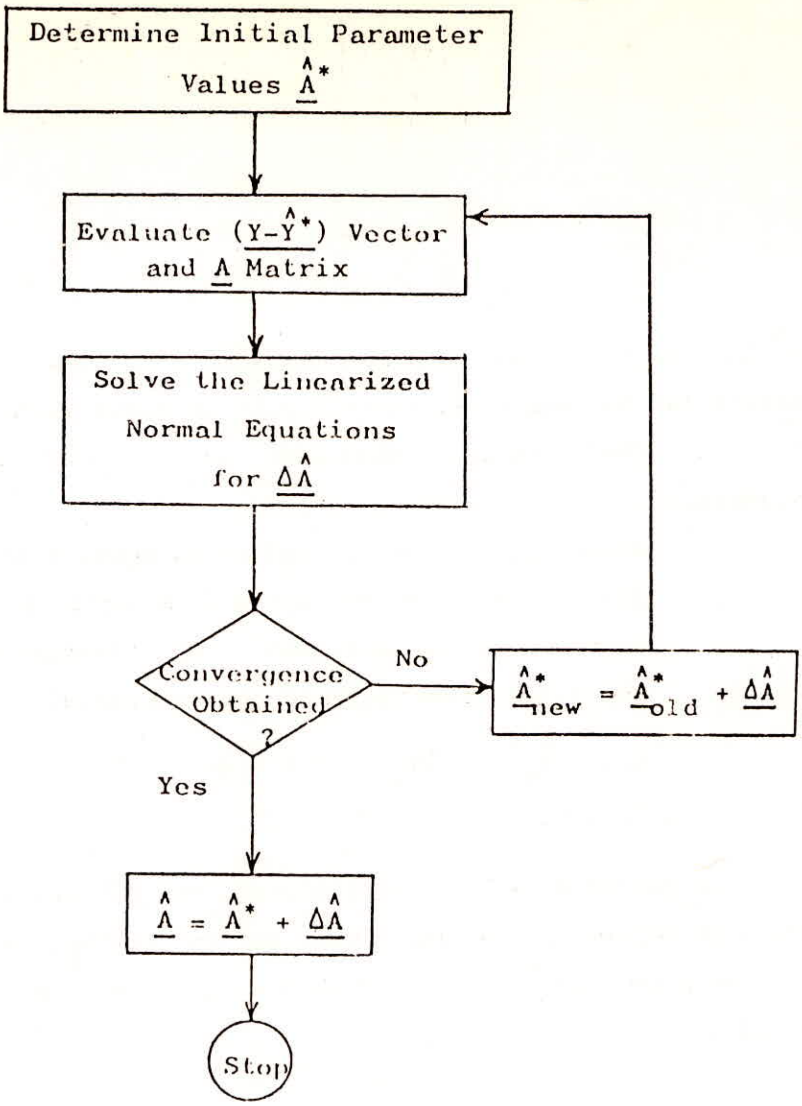

Figure 13 . Gauss Newton (BARD ALGORITHM) Logic Diagram

converge rapidly for good starting estimates. Thus in the Marquardt procedure, the initial values of λ are large and will decrease toward zero as the optimum is approached. All other computational details parallel the Gauss Newton procedure. The above procedure is illustrated through a flow chart in Fig.14.

5.0 HYDROLOGICAL MODEL PARAMETERS OPTIMISATION

The selection of a parameter optimisation algorithm for solving the non-linear programming problems particularly in the area of hydrological modeling is complicated by the decisions taken about the type of model, the selected model functions and their structure, the time interval and the length of historic record available and its quality. The routine application of a particular technique can lead to a needless precision in the values obtained for parameters and consequent waste of computing resources, or if the model parameters have a degree of dependence then the operation of the model structure can be badly affected. In this case although a good fit has been apparently obtained in the calibration period the effect on the model structure of badly skewed parameters can lead to failure in prediction mode even though the concepts used in the expressions are valid; their rethinking would have little effect.

The array of search techniques in use range from the direct application of the experience and subjective judgments of the modeler on a 'trial-and-error' basis using field parameters spatially averaged over the whole basin to the use of automatic search techniques. The latter should perhaps be called 'semi automatic' since they can not be applied without experienced analysis of the results. All the search techniques including Rosenbrock optimisation technique are a form of direct search in which a parameter is changed and the result tested against observed data and previous model outputs. The variations occur in the strategy adopted in changing parameters and testing the results from the model. The only significant variations from this are the derivative based techniques where the matrices of first

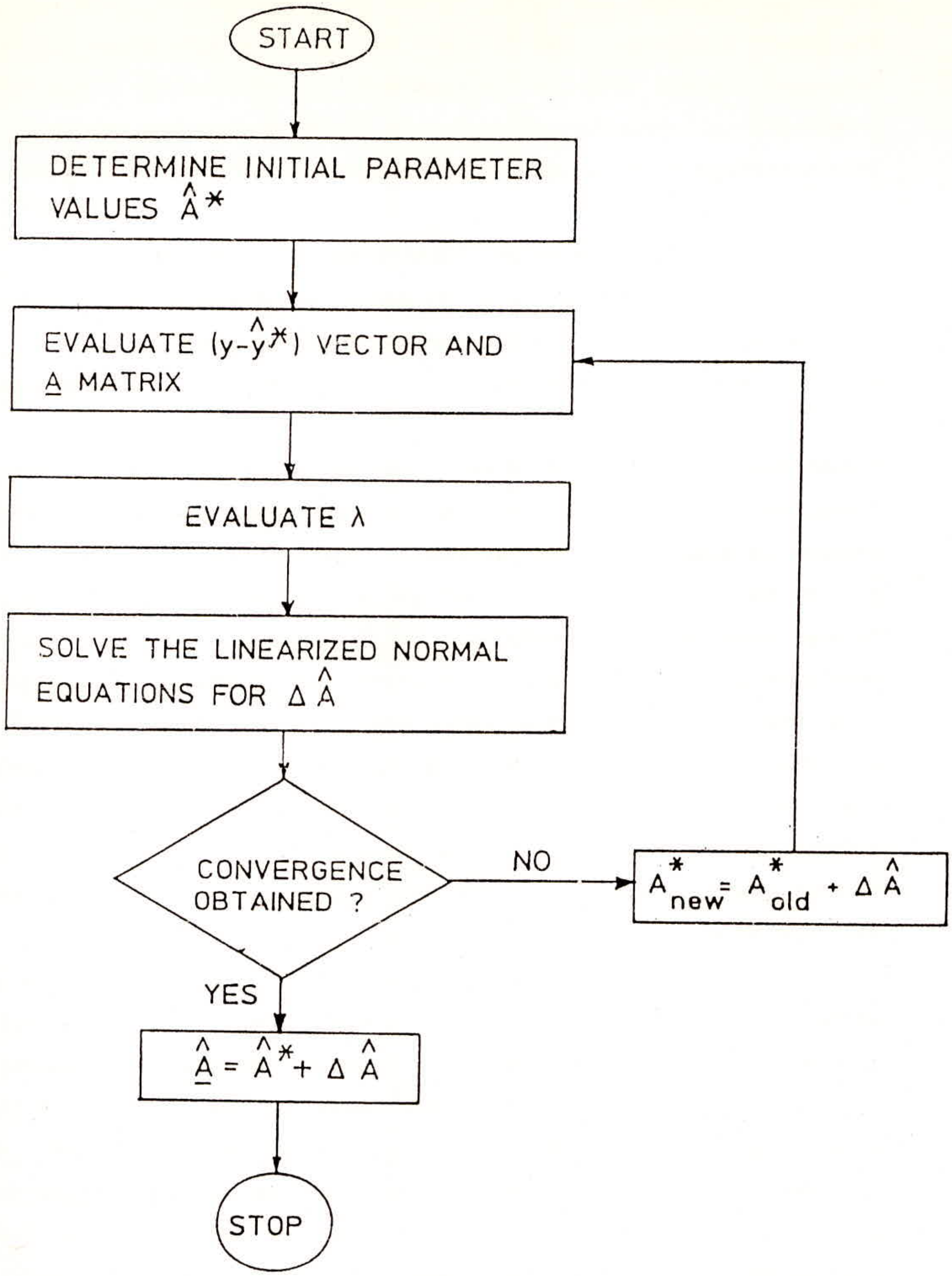

FIG. 14 FLOW CHART ILLUSTRATING MARQUARDT ALGORRITHM

and second derivatives of the gradient vectors are considered to obtain a Hessian, confirming that a minimum has been reached in the hyperspace generated by the parameters. With data derived from a function that is smoothly analytic this is the probably the

best type of algorithm to use and convergence can be extremely rapid. However, the position is not so simple with a data set containing errors and possible short term bias. This can affect the Jacobian of gradient Vectors Very Significantly, and the consequent ill conditioning of the set of parametric Vector equations must lead of failure in obtaining a unique set of optimal parameters. The problems of using these derivative based algorithms on observed data have not been investigated thoroughly, though there are models which are designed to avoid the problems (Marquardt, 1963).

With the direct search techniques the success of the variation of a parameter can be assessed either by direct comparison of plots of the observed with the model predicted hydrograph subjectively or by the use of some objective function based on the residual errors between the two sets of flows. This function assumes particular importance with an automatic search algorithm as it generates the parametric hyperspace in which the search takes place for the minimum, the hyperspace having the same number of dimensions as there are parameters. The most commonly used direct search technique is the Rosenbrock technique which is very sensitive to gradients in the hyperspace topography and quickly finds the nearest local minimum. At convergence the optimum obtained has to be tested in an endeavour to find out if it is a local or global optimum, and one satisfactory way of achieving this was found to be restart the algorithm iterations with a new set of parameters. By this repeat of the process from a new starting point a larger area around the optimum is searched.

REFERENCES

1. Fletcher, R. (1980),'Practical Methods of Optimisation', Vol.1, John Wiley & Sons.

2. Himmelblau, D.M. (1972),'Applied Non-Linear Programming, McGraw Hill, New York.

3. Jacoby, S.L.S., J.S. Kowalik, and J.T. Pizzo (1972), 'Iterative methods for Non-Linear Optimisation Problems', Prentice Hall, Inc., Englewood cliffs, N.J.

4. Marquardt, D.M. (1963),'An Algorithm for Least Squares Estimation of Non-Linear Parameters', J. of Soc. Indus. App. Math, 11, pp.431-441.

5. Nash, J.C. (1984),'Compact Numerical Methods for Computers, Linear Algebra and function minimisation', Adam Hilger Ltd., Bristol.

6. Rosenbrock, H.H. (1960),'An Automatic Method for Finding the Greater or Least Value of a function', Computer J., Vol.3, pp.175-184.