STUDIES ON NONLINEAR OPTIMIZERS

A DISSERTATION

submitted in partial fulfilment of the
requirements for the award of the degree
of
MASTER OF ENGINEERING
in
CHEMICAL ENGINEERING
With Specialization in
COMPUTER AIDED PROCESS PLANT DESIGN

By

SANJEEV SINGH

DEPARTMENT OF CHEMICAL ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE-247 667 (INDIA)

JUNE, 1990

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in this dissertation entiled " **STUDIES ON NONLINEAR OPTIMIZERS** " in partial fulfilment of the requirements for the award of the degree of **MASTER OF ENGINEERING** with specialization in **COMPUTER AIDED PROCESS PLANT DESIGN**, submitted in the department of Chemical Engineering, University of Roorkee, Roorkee is an authentic record of my own work carried out for a period of about ten months, from Aug. 1989 to May 1990 under supervision of Dr. A.K. AGARWAL, Mr. S.N. SINHA, Mr. R.BHARGAVA, Department of Chemical Engineering, University of Roorkee, Roorkee, India.

The matter embodied in this dissertation has not been submitted by me for the award of any other degree.

Sanjeev Singh

Dated : June 2 ,1990 **(SANJEEV SINGH)**

This is to certify that above statement made by the candidate is correct to the best of our knowledge.

Agarwal

(Dr. A.K. AGARWAL)
Lecturer
Chemical Engg.Deptt.
University of Roorkee,
ROORKEE - 247 667

(Mr. S.N. SINHA)
Lecturer
Chemical Engg. Deptt.
University of Roorkee,
ROORKEE - 247 667

RBhargava

(Mr. R. BHARGAVA)
Lecturer
Chemical Engg. Deptt.
University of Roorkee,
ROORKEE - 247 667

ACKNOWLEDGEMENT

I wish to express my appreciation to all who provided assistance during the course of this thesis work. The following deserve special mention :

Dr. A. K. Agarwal, Mr. R. Bhargava and Mr. S. N. Sinha , for their supervision and inspiring guidance throught the work.

Professor & Head of the Department, Dr. B. S. Varshney for his constant inspiration during the work.

The technical staff of the Department of Chemical Engineering , for their Co-operation during the work.

Finally all my friends, for their necessary and valuable assistance during the thesis work.

Sanjeev Singh

(SANJEEV SINGH)

DEDICATED TO

MY PARENTS

ABSTRACT

In the area of chemical engineering, two aspects are very common. Firstly, expressing the behaviour of a system in a mathematical form (that is,model) and secondly the system performance is optimized for its known variables. For both the cases, one need to have optimizer which could minimize or maximize the function for these two aspects.

The object of the present study was to provide the FORTRAN code of different optimizers (mainly non-linear optimizers) to facilitate the department's software facilities.

Three optimizers were selected namely, Complex method,Interior penalty function method and non-linear least square Marquardt method. Two objective functions in the area of chemical reaction engineering field namely, parameter estimation and sequential experimental design,were utilized to check their functioning and compare performance.

Marquardt method showed superiority over Complex method for parameter estimation problem. Also, Marquardt method showed that it can utilize the worst starting point.

However, Complex method could be used for design of experiment in comparison to Interior penalty function method. Also, Complex method showed a tendency to reach in vicinity of optimum solution in for fewer function evaluations. This behavior could be utilized to initiate the optimization and could be ovtaken the superior optimizer to achieve the optimum solution.

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

n	number of independent variables
X	vector of independent variables
x_i	ith component of X
$f(X)$	objective function
m	total number ofconstraints
$G_j(X)$	jth constraint
R_i	random number corresponding to ith valable
l_i	lower limit on ith variable
u_i	upper limit on ith variable
P_h	worst point
P_o	reflection of the worst point
$\bar{P}$	centroid of all points excluding P_h
$\in$	preset tolerance
Y_i	value of function corresponding ith vertice
k	number of vertices
$\nabla f(X)$	qradient vector
S_i	search direction in ith iteration
λ_i^*	optimum step length in ith iteration
H_i	Hessian matrix in ith iteration
$M(X)$	least square function
$J(X_k)$	Jacobian matrix in kth iteration
ΔX_k	correction vectrorin kth iteration
I	Identity matrix
ϕ	penalty function
r_k	penalty parameter
C	panalty reduction parameter

$E(y)$	expected value of dependent variable y.
β	$[\beta_i, \beta_2, \ldots \beta_k]$ parameters
$r_{C_2H_6}$	rate of formation of ethane kgmol/kgcatalyst)
k	reaction rate constant (kgcatalyst) second/kgmol
A_o	preexponential factor
E	Activation energy Kcal /kgmol
T	Absolute temperature.
K	vector of model parameter
R	ideal gas constant
p	number of parameters
D	(nxp) matrix of element $\partial f(X_u, K) / \partial K_j$
$X_{C_2H_4}$	mole fraction of ethylene
X_{H_2}	mole fraction of hydrogen
y_u	measured reaction rate for experiment u
α	order of reaction w.r. to ethylene
β	order of reaction w,r to hydrogen
σ^2	variance of random error
μ	mean
ϵ_u	random error

CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

Chemical engineering is vast field and its two areas modelling and system optimization proves to be very vital. Modelling is area where one expresses the behavior of the system in mathematical form. Sometimes these models are utilized. To estimate constants appearing in a model and in system optimization to achieve better system performance, one has to have an optimizer which minimizes or maximizes the function for the both cases.

During last two decade, the use of digital computer for real time data processing has formed wide application, both in research and industrial environment.The advent of high speed digital computation has also increased the application of optimization techniques. The significance of optimization techniques in this context lies in attempting to adjust some system's variables to improve the performance of that system, measured in some way.

There is no single technique avalaible for solving all optimization problems efficiently. Hence a number of optimization techniques have been developed for solving different type of optimization problems. a voluminious bundle of mathematical optimization techniques has evolved to take advantage of immense capabilities of digital computers.

Models expressing the behavior of system and the functions expressing the system performance are ,in general, non linear in nature. Non linear optimization techniques are ,therefore, employed in such situations. Many nonlinear optimization algorithms have been suggested to solve these problems, but far more algorithms have been proposed than have been successfully applied. The range of applicability of existing non linear optimizers is ,therefore, limited.

As implied above , the aim of present study was to test the functioning and performance of few non linear optimizers, for one or more, known objective functions.

Three non linear optimizers, namely Complex method , Interior penalty function method and non linear least square Marquqrdt method were proposed to be studied in detail. These optimizers are discussed with their algorithms in Chapter 3. The functioning and performance of these optimizers were tes two objective functions dicussed in Chapter 4.

CHAPTER 2

NONLINEAR PROGRAMMING

CHAPTER - 2

NONLINEAR PROGRAMMING

2.1 INTRODUCTION

Non linear programming deals with the optimization of the nonlinear function subject to linear or nonlinear constraints. The function to be minimised or maximised is called objective function. Objective function represents cost, throughput or the like. Methods (algorithms) of solving nonlinear programming problem are known as nonlinear optimizers.

Quite a large number of algorithms have been proposed to solve the general nonlinear programming problem, mainly because of, no one method appears to be far superior to all the others. Nonlinear programming problems are divided into two categories, unconstrained and constrained problems.

2.2 UNCONSTRAINED NONLINEAR METHODS

Seveal methods are available for solving an unconstrained minimization problem. These methods can be classified into two broad categories as direct search methods and descent methods. These methods are shown in table 2.1.

TABLE 2.1

UNCONSTRAINED MINIMIZATION METHODS

Direct search methods	Descent methods
(i) Random search	(i) Steepest-descent method
(ii) Univariate search	(ii) Conjugate gradient
(iii) Pattern search method	(iii) Newton's method
(a) Powell's method	(iv) Variable metric method
(b) Hook and Jeeve's method	(Davidon Fletcher-Powell)
(iv) Rosenbrock's method of rotating coordinate	
(v) Simplex method	

2.2.1 DIRECT SEARCH METHODS

The direct search methods require only objective function evaluations and do not use the partial derivatives of the function in finding the minimum and hence are often called non gradient methods. These methods are most suitable for simple problem involving a relatively small number of variables.Some of the direct search method are being discussed here

2.2.1.1 RANDOM SEARCH METHODS

The random search methods are based on the use of random numbers in finding the minimum point, these methods can be used quite conveniently. Brief discussion of method is as follows:

Let the problem to find the minimum of f(X) in n dimensional space defined by

$$l_i \leq x_i \leq u_i \quad i = 1.2 \ldots.n \quad \text{. . . . .}(2.1)$$

where l_i, u_i are the lower and upper bounds on the variables x_i. In this method, sets of n random numbers $(r_1, r_2, r_3 \ldots\ldots r_n)$ are generated that are distributed between 0 and 1. Each of set of these number is used to find a point, X, inside the space defined by equation (2.1) as

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} l_1 + r_1 & (u_1 & - & l_1) \\ l_2 + r_2 & (u_2 & - & l_2) \\ \vdots & \vdots & & \vdots \\ l_n + r_n & (u_n & - & l_n) \end{bmatrix} \ldots\ldots\ldots \quad (2.2)$$

and the value of function is evaluated at this point X by generating a large number of points and evaluated the value of the objective function at each of these points,least value of the f(X) is selected as desired optimum . These methods posses the following advantages .

(1) These methods can be worth even if objective function is discontinuous and non differentiable at some of points.

(2) These methods can be used to find the global minimum when the objective function posseses several local minima.

(3) These methods are applicable when the other methods fail due to local difficulties such as sharply varying functions and shallow regions.

(4) Although these methods are not very efficient by themselves, they can be used in the early stage of optimization to detect the region when global minima is likely to be found. Once this region is found, some of more efficient techniques can be used to find the precise location of global minimum point.

2.2.1.2 UNIVARIATE METHOD

In this method only one variable is changed at a time, and remaining points are best constant so that it can produce a sequence of improved approximations to the minimum points. It takes the fixed values of n-1 variables at a base point X_i in the i_{th} iteration and vary the remaining variable. Since only one variable is changed the problem becomes one dimensional minimization problem and any of one dimensional method can be used to produce a new base point X_{i+1}. The search is now continued in a new direction. This new direction is obtained by changing any one of the n-1 variables that were fixed in previous iteration. In fact, the search procedure is continued by taking each coordinate direction in turn. After all the n directions are searched sequentially, the first cycle is completed and hence the entire process of sequential minimization is repeated. This procedure is continued untill no further improvement is possible in the objective function in any of n directions of a cycle.

Theoretically this method can be applied to find the minimum of any function that possesses the continuous derivatives. However, if the function has a steep valley , the method may not even converge.

2.2.1.3 PATTERN SEARCH METHOD

Each step of pattern search method comprises two kinds of moves: exploratory and pattern. In univariate method only exploratory moves are performed by changing only one variable at a time, a procedure known to be often very inefficient when applied to optimization problems. For this reason Hook and

Jeeves(1961) suggested that the exploratory moves be followed by pattern moves and direction of search is determined from result of exploratory move. The search begins at an arbitrary chosen starting point and with prescribed step length λ_i > 0 in each of the coordinate directions x_i , i = 1,2.....n

Two well known pattern search methods are Hook and Jeeves's (1961) method and Powell's (1964) method.

2.2.1.4 SIMPLEX METHOD

The simplex method is based on the comparison of the objective function values at (n + 1) vertices of a simplex. A set of (n + 1) equidistant points in n-dimensional space forms a regular simplex .This simplex moves towards the optimum point (Nelder-Mead 1965). This movement is achieved by three basic operations: reflection, contraction and expansion.

REFLECTION

Initially and at times during convergence, a new point is generated that is a reflection of the worst point through the centroid of all other points. If the reflection point is better than the current best point, it replaces the current worst point and an expansion step is tried. If the reflection point is worse than the current best point, the expansion is bypassed and reflected point is compared with all other points except the worst. If the reflection point is better, it replaces the worst point.

The reflection of the worst point, P_h is denoted by P^*, and its coordinates are defined by relation

$$P^* = (1 + \alpha) \bar{P} - \alpha P_h \dots\dots\dots\dots\dots (2.3)$$

where $\bar{P}$ is the centroid of all points excluding P_h and α is a positive constant, the reflection coefficient . Thus P^* lies on the line joining P_h and P, but on the far side of new simplex.

CONTRACTION

If the reflection is worse than all points except the current worst point, a contraction is generated by selecting a point between the worst point and the centroid. This second trial at a new point P^{**} is defined by,

$$P^{**} = \beta P_h + (1 - \beta) \bar{P} \dots\dots\dots\dots\dots\dots\dots\dots\dots (2.4)$$

where β the contraction coefficient lies between o and 1. If the contraction fails to improve the situation the size of simplex is reduced by moving all the points half the distance toward the best point.

EXPANSION

If the reflection point is better than the current best point, an expansion step is tried .This is done by extending the reflection point in the direction between it and the centroid. This expansion point P_{oo} is defined by relation

$$P_{oo} = P^{**} + (1 - \tau) \bar{P} \dots\dots\dots\dots\dots\dots\dots (2.5)$$

where τ is the expansion coefficient and is greater than unity. If the expansion point is successful, it replaces the reflection point, otherwise the reflection point remains in the simplex. In either instance the simplex procedure is restarted.

The programmed procedure is terminated when following condition does satisfy.

$$[\sum_{i=1}^{n} (y_i - \bar{y})^2 / n]^{.5} \leq \epsilon$$

where ϵ = Preset value greater than zero

$$\text{and} \quad \bar{y} = \sum_{i=1}^{n} y_i / n \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.6)$$

Nelder and Mead found that useful values for coefficients α,β and τ were : $\alpha = 1$, $\tau = 2$, $\beta = 0.5$

The simplex method has a wide application since it does not make any assumtions about the objective function except that the function is continuous . It may be useful when the locations of minima are needed with a limited accuracy and final rapid convergence is not essential. Some disadvantages of method are:

(1) There is a possibility that the simplex may collapse into a subspace and will not find the solution in the desirable space of all the variables.

(2) It can also shrink drastically in a steep valley and stop the procedure prematurely.

(3) Computational efficiency depends upon choice of coefficient α,β and τ and on the size of the initial simplex.

(4) No direct use is made of the information acquired in previous steps.

2.2.2 DESCENT METHODS

The descent techniques require, in addition to function to evaluations, the evaluation of first and second derivatives of the objective function. Since more informations about the function being minimized is used,the descent methods are generally more efficient compared to direct search methods.The descent methods are also known as gradient methods. Some of the descent methods have been discussed in detail in following sections:

2.2.2.1 STEEPEST - DESCENT METHOD

In this method negative ofgradient vector is used as a direction for minimization. This idea was first introduced by Cauchy (1847). In these methods a trial point (X_i) is selected and optimum point is sought by the iterative procedure using following equation.

$$X_{i+1} = X_i + \lambda_i^* S_i \ldots\ldots\ldots\ldots\ldots (2.7)$$

$$S_i = -\nabla f_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.8)$$

where $\lambda_i^*$is optimum step length along search dirction $S_i = -\nabla f_i$. It can be determined using any one dimensional method of minimization. Process is terminated if the components of search direction are less than preset small quantity $\epsilon > 0$.

The steepest descent technique is a simple technique involving relatively little computational work per step. Each iteration is independent of the previous iteration, hence method is relatively economical in computer storage. The method may converge to a solution even from a poor strating point. Unfortunately method has several short comings. Very short step may be produced accompained by sharp changes in the gradient f(x). This is frequently refered to as "zigzagging" and implies, of course, slow convergence and low computational efficiency inspite of the small amount of work per step. Furthermore steepest descent technique depends on the variable scaling (to scale the original variable used in method).

In conclusion it is recommended that steepest descent method be used only as an auxiliary technique in combination with other minimization technique.

2.2.2.2 CONJUGATE GRADIENT METHOD

The convergence characteristic of the steepest descent method can be greatly improved by modifying it in to a conjugate gradient method. It has shown that any minimization method that makes use of the conjugate direction is quadratically convergent. This property of the quadratic conguence is very useful because it ensures that the method will minimize a quadratic function in n steps or less. Since any general function can be approximated reasonably well by a quadratic near the optimum point, any quadratically convergent method is expected to find the optimum point in a fixed number of iterations. This method was first suggested by Fletcher and Reeves (1964).

The conjugate gradient methods have the properties of first order gradient combined with quadratic convergence due to its conjugacy of the descent directions. They have overall good convergence properties and the Fletcher-Reeves method has also a relatively modest requirement for computer storage. The computer codes corresponding to the methods are simple. Normally, more than one cycle of n steps is required, and in resetting the computation all information collected in previous iterations is discarded. The use of this mehod in preference to the variable metric technique (discussed next subsection) is recommended only if computer storage economy is more important than computational efficiency. This method should be used however,in preference to the first order gradient order method.

2.2.2.3. VARIABLE METRIC METHOD

A significant developement in area of descent techniques was made by Daviden [1959] and extended by Fletcher - Powell [1963] this work led to the so called " Variable metric " method of minimization.

The variable metric typically has the following algorithm structure.

(1) Given x_1 and positive definit matrix H_1 (Initially H_1 = I, n dimensional identity matrix) set k=1 and compute gradient vector $\nabla f(x_1)$

(2) Calculat the search direction S_k using following formula

$S_k = - H_k^T \ \nabla f(X_k)$................. (2.9)

(3) Find the optimal step lenth λ_k^* in the direction S_k and set

$X_{k+1} = X_k + \lambda_k^* S_k$

(4) Test the new point X_{k+1} for optimality. If X_{k+1} is

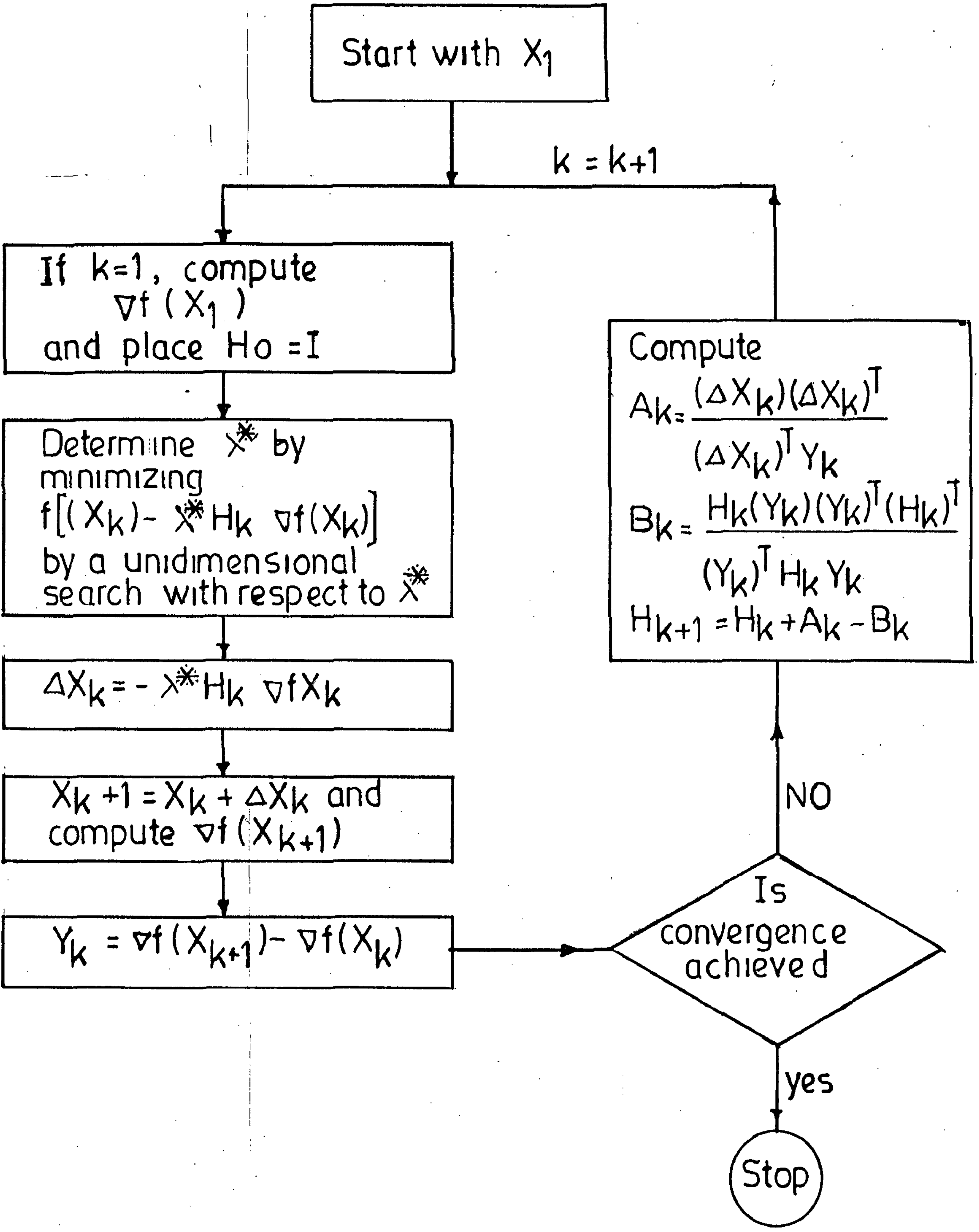

Start with X_1
$k = k+1$
If $k=1$, compute $\nabla f(X_1)$ and place $H_0 = I$
Determine λ^* by minimizing $f[(X_k) - \lambda^* H_k \nabla f(X_k)]$ by a unidimensional search with respect to λ^*
$\Delta X_k = -\lambda^* H_k \nabla f X_k$
$X_k + 1 = X_k + \Delta X_k$ and compute $\nabla f(X_{k+1})$
$Y_k = \nabla f(X_{k+1}) - \nabla f(X_k)$
Is convergence achieved
NO
yes
Stop
Compute
$A_k = \frac{(\Delta X_k)(\Delta X_k)^T}{(\Delta X_k)^T Y_k}$
$B_k = \frac{H_k(Y_k)(Y_k)^T(H_k)^T}{(Y_k)^T H_k Y_k}$
$H_{k+1} = H_k + A_k - B_k$

Fig. 2·1. Flow chart for variable metric method optimal terminate the procedure, otherwise go to step (5)

(5) Update the H matrix as

$$H_{k+1} = H_k + M_k - N_k$$

Where
$$M_k = \frac{S_k \; S_k^T}{S_K^T \; Q_k}$$

$$N_k = \frac{(H_k Q_k)\;(H_k Q_k)^T}{Q_k^T \; H_k \; Q_k}$$

and
$$Q_k = \nabla f(X_{k+1}) - \nabla f(X_k)$$

(6) Set the new iteration number k = k+1 and go to step (2)

The Flow chart for the method is shown in figure 2.1

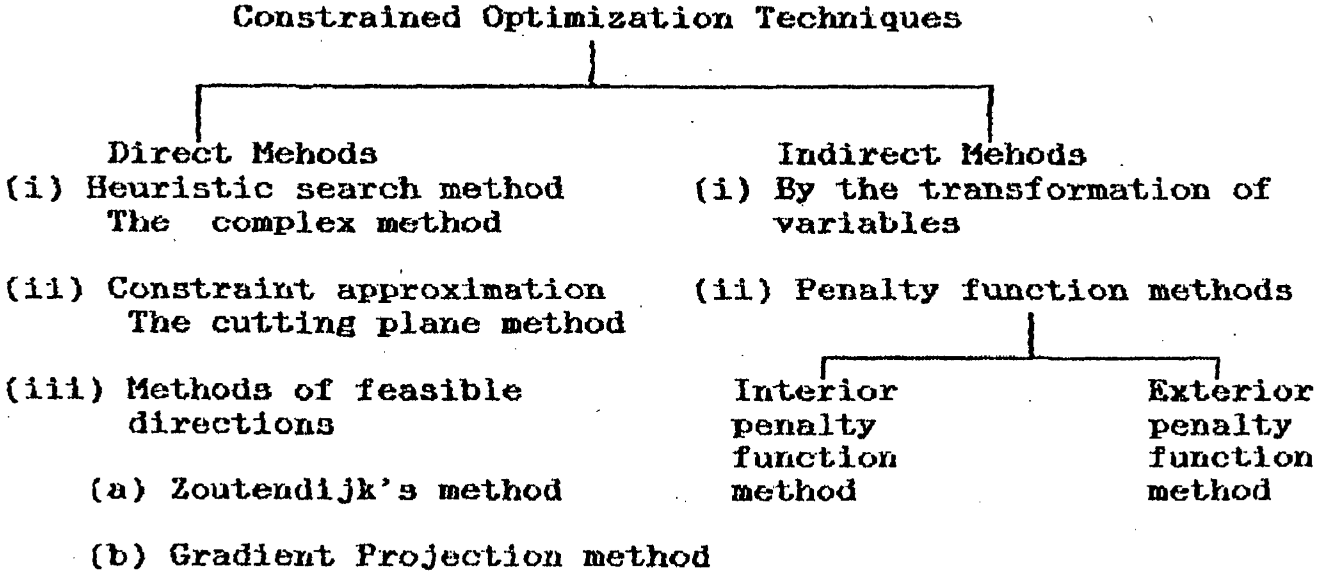
The variable metric techniques have many of advantages without having some of their disadvantages. The advantages include fast convergence near a minimum and fitness for a quadratic function. Due to symmetry of H_k, the methods are also comparable in their computer storagee requirements. Comparison with other quadratically convergent methods show that variable metric methods perform better for general non quadratic functions. Thus quadratic convergence alone does not indicate the full strength of a method. It is likely that fast convergence of the method when applied to a general nonquadratic function due to fact that H_k approximately satisfies $H_k \; Q_{k-1} = \Delta x_{k-1}$ which is analogus to $A^{-1} \; Q_{k-1} = \Delta X_{k-1}$ for quadratic functions.

The variable metric methods avoid the computation of second order derivaties for the objective function. The methods are basically stable. They generate directions of descent because the matrix Hk is positive definite.

2.3 CONSTRAINED METHODS:

There are many techniques available for the solution of a consrained nonlinear programming problem. All these methods can be classified into two broad categories, namely direct methods and indirect methods as shown in table 2.2.

TABLE 2.2



Constrained Optimization Techniques

Direct Mehods
(i) Heuristic search method
The complex method
(ii) Constraint approximation
The cutting plane method
(iii) Methods of feasible directions
(a) Zoutendijk's method
(b) Gradient Projection method

Indirect Mehods
(i) By the transformation of variables
(ii) Penalty function methods
Interior penalty function method
Exterior penalty function method

2.3.1 DIRECT SEARCH METHODS

In the direct search methods, the constraints are handled in explicit manner .Brief discussion of all three categories of direct search method is being discussed here.

2.3.1.1 HEURISTIC SEARCH METHODS

The heuristic search methods are mostly intutive and do no have much theortical support. Complex method is based on th heuristic search. Detailed discription of the complex method ha been given in next chapter.

2.3.1.2 CONSTRAINT APPROXIMATION METHODS

In these methods, the nonlinear objective function and the constraints are linearized about some point and approximating linear programming problem is solved by using linear programming technique (L P) (simplex method).

The resulting solution is then used to construct a new linear approximation which will be solved by using LP techniques. This procedure is continued untill the specified convergence criteria is satisfied. There are two methods, namely the cutting plane method and approximate programming method which work on this principle.

Methods have following advantages and disadvantages.

ADVANTAGES:

1. Direct extension of the simplex method (linear programming) therefore efficient for convex programs which are nearly linear.
2. Relatively little work per step.
3. Simple computer program.
4. Some problems with an infinite number of constraints can be solved.

DISADVANTAGES:

1. Methods can not be applied to concave problems.
2. Intermediate solutions are not feasible.
3. Rather slow convergence, especially if the minimum is not

in a vertex.

4. Linear subprograms will consist of near dependent constrains which could lead to serious rounding-off errors, especially if the minnnimum is not in a vertex.
5. Rather inefficient for problems with linear and a nonlinear possibly quadratic, objective function.

2.3.1.3 METHODS OF FEASIBLE DIRECTION

The methods of feasible directions are those which produce an improving successive of feasible vector xi , by moving in a succession usable feasible directions. A feasible direction is one along which at least a small step can be taken without leaving the feasible domain.A usable feasible direction is a feasible direction along which the objective function value can be reduced at least by a small amount.

Each iteration consists of two important steps in the methods of feasible directions. The first step consists of finding a usable feasible direction at a specified point and second step consist of determining a proper step length along usable feasible direction found in the first step. The Zoutendijk's method of feasible directions and Rogen's gradient projection method are considered as particular cases of general method of feasible directions.

Feasible directions methods offer an efficient way of reducing the nonlinear programming problem to a sequence of linear programming problems (if a linear normalization is used). From pratical point of view, they offer the additional advantages of providing feasible intermediate approximations to the solution. The methods have been implemented to solve practical problems of considerable size and to solve practical problems of considerable size,and the methods proved to be successful. The following are disadvantages of the methods.

(1) A rather complicated computer program.

(2) Inability to handle the equality constraints without special devices.

(3) Determining of the step length needed, resulting in more work per step.

2.3.2 INDIRECT SEARCH METHODS

In indirect search method the constrained problem is solved as a sequence of unconstrained problem by incorporating constraints into objective function. Different methods of this category is discussed below:

2.3.2.1 TRANSFORMATION OF VARIABLES

Some of the constrained optimization problems have their constraints expressed as simple and explicit functions of the decision variables. In such a cases it may be possible to make a change of variables such that constrainst are automatically satisfied. In some other cases, it may be possible to know in advance which constraints will be active at the optimum solution. In these cases,the information that, paticular constraint equation $g_i(x)=0$,can be used to eliminate some of the variables from the problem.

Use of transformation of variable is that the solution of constrained problems is eased considerably by eliminating the constraints. So that one of the more powerful methods of for unconstrained optimization, e.g. that of Powell (1964), may be employed.

The transformation have successfully been used again in conjuction with Powell's method for problems with up to twenty independent variables. These problems arose from a " design of experiments" study . The aim in each case was to minimise the size of confidence region of the estimates of parameters in the assumed model.

2.3.2.2 PENALTY FUNCTION METHODS

Fiacco and McCormic (1968) have developed the penalty function method for solving constrained optimization problems. In this approach a penalty term reflecting the constraints violations multiplied by a scalar weight is augmented to the actual objective function. If objective function is minimized for a sequence fo decreasing penalty weights, the solution of the successive unconstrained problems approach the constrained solution.

There are two type of penalty function methods the interior penalty function method and the exterior penalty function method. In both type of methods, the constrained problem is transformed into a sequence of unconstrained minimization problems such as that constrained minimum can be obtained by solving the sequence of unconstrained minimization problems.

In the interior penalty function methods, the sequence of unconstrained minima lie in the feasible region such that it converges to constrained minima from the interior of feasible region. In the exterior methods, the sequence of unconstrained minima lie in the infeasible region and converges to the desired solution from the exterior of the feasible region.

Methods have following advantages and disadvantages.

ADVANTAGES

(1) Applicable to the nonconvex problems including those with nonlinear constraints.

(2) Very efficient for unconstrained problems as well as for the problems with a few highly nonlinear constraints.

(3) Good convergence can be expected if penalty parameter (r_k) are well choosen and an extrapolation device is used.

(4) Intermediate solutions feasible.

(5) Relatively simple computer program.

DISADVANTAGES:

(1) A special structure of the constraints are (linearity or near linearity or partical linearity) destroyed, even constraints like $X_i \geq 0$ are not dealt with in a special simple way.

(2) Much work per step.

(3) Rounding off problems may some times arises as penalty parameter tends to zero and simultaneously constraints tends to infinity. Practical experiments are very promising in this respect, however.

(4) No upper bound for the value of objective function available.

(5) Problems with an infinite number of constraints can not be solved with the methods in their present form.

2.4 ONE DIMENSIONAL MINIMIZATION METHODS

The aim of one dimensional methods is to find optimum step length in current search direction in multivariables problem. several one dimensional minimization methods are available, which are given below.

(1) Dichotomous (2) Fibonacci method (3) Golden section

(4) Interpolation methods

Interpolation methods divided into two categories ,requiring no derivatives (Quadratic) and requiring derivatives (Cubic).

Interpolation methods were originally developed as one dimensional search within multivariable optimization technique, and are generally more efficient compared to Fibonacci type approaches. For present study cubic interpolation method has been selected. Iterative steps of method are given below

(1) start with A_0 and find a point λ= B at which slope $df/d\lambda$ is positive, where $df/d\lambda = f' = df/d\lambda(X+\lambda S) = S^T \nabla f(X+\lambda S)$. The point B can be can be taken as first value out of t_0, $2t_0$, $4t_0$..... at which f' in nonnegative , where t_0 is a preassigned initial step size.

(2) Cubic equation $h(\lambda) = a + b\lambda + c\lambda^2 + d\lambda^3$ is used to approximate the function $f(\lambda)$ between A and B. find the value of f_A, f_B, f'_A, f'_B in order to calculate a,b,c and d in above equation.

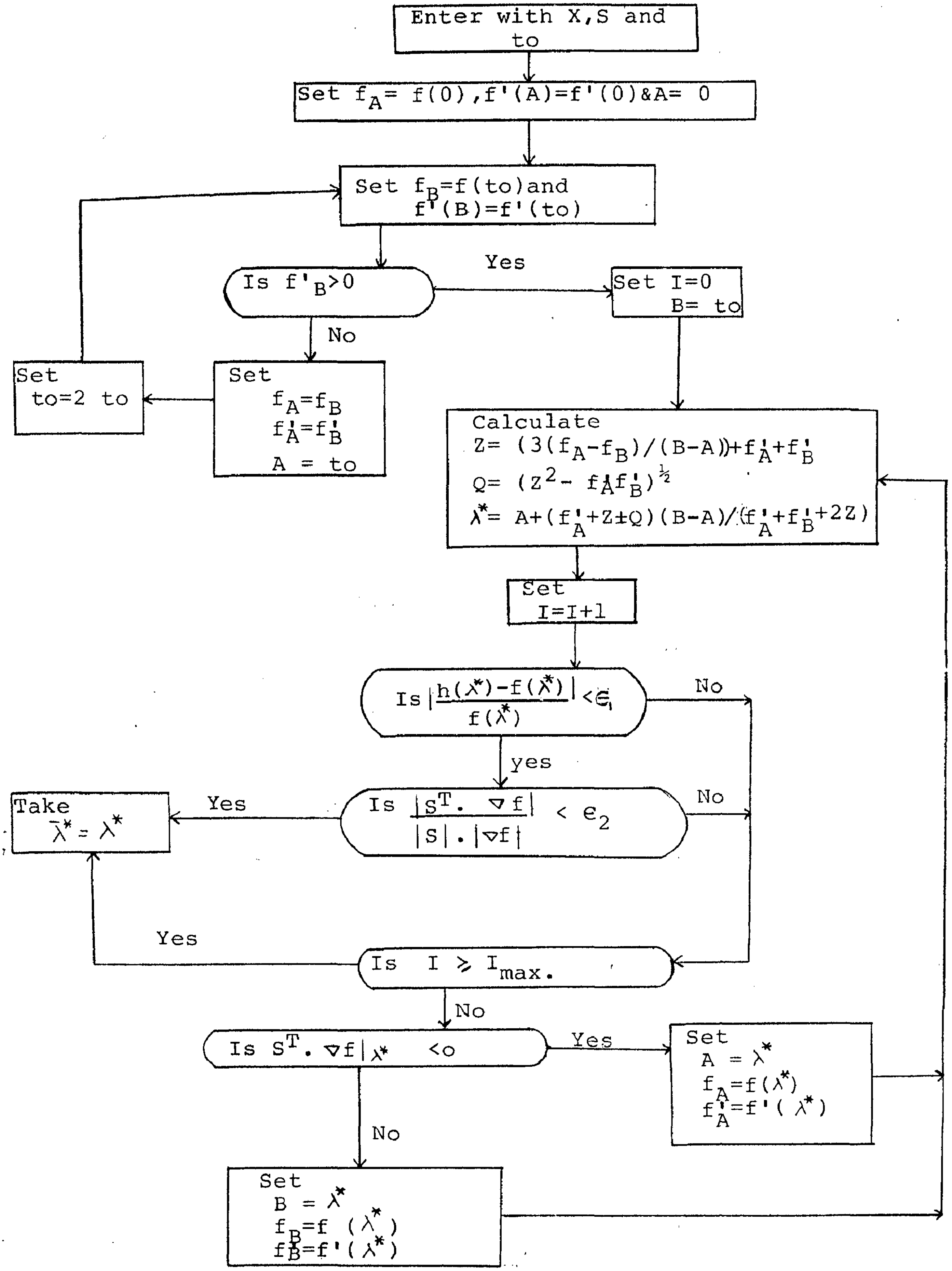

Enter with X,S and to
Set f_A= f(0),f'(A)=f'(0)&A= 0
Set f_B=f(to)and f'(B)=f'(to)
Is f'_B>0
Yes
Set I=0 B= to
No
Set $f_A=f_B$ $f'_A=f'_B$ A = to
Set to=2 to
Calculate
$Z= (3(f_A-f_B)/(B-A))+f'_A+f'_B$
$Q= (Z^2- f'_A f'_B)^{\frac{1}{2}}$
$\lambda^*= A+(f'_A+Z\pm Q)(B-A)/(f'_A+f'_B+2Z)$
Set I=I+1
Is $\left|\frac{h(\lambda^*)-f(\lambda^*)}{f(\lambda^*)}\right|<\epsilon_1$
No
yes
Is $\frac{|S^T. \nabla f|}{|S|.|\nabla f|} < \epsilon_2$
No
Yes
Take $\bar{\lambda}^* = \lambda^*$
Yes
Is I $\geqslant$ $I_{max.}$
No
Is $S^T. \nabla f|_{\lambda^*}$ <o
Yes
Set A = λ^* $f_A=f(\lambda^*)$ $f'_A=f'(\lambda^*)$
No
Set B = λ^* f_B=f (λ^*) $f'_B=f'(\lambda^*)$

Fig.2.2 Flow chart for cubic interpolation method.

$$b = [\ B^2 f'_A + A^2\ f'_B + 2\ A\ B\ Z]/(A-B)_2$$

$$c = -\ [\ (\ A+B\ )Z + B\ f'_A + Af'_B\ ]/(A-B)_2$$

$$d = [\ 2\ Z + f'_A + f'_B\ ]/3(A-B)_2$$

where $Z = -3(\ f_A - f_B)/\ (B-A) + f'_A + f'_B$

(3) Calculate $\tilde{\lambda}^*$ using following eqation.

$$\tilde{\lambda}^* = A + \ (f'_A + Z \pm Q\ )(B-A)/(\ f'_A + f'_B + 2\ Z)$$

where $Q = (Z^2 - f'_A f'_B)^{1/2}$

(4) Check convergence. Following convergence criteria can be used.

$$\left| (h(\tilde{\lambda}^*) - f(\tilde{\lambda}^*))/f(\tilde{\lambda}^*) \right| < \epsilon_1$$

Flow chart for implementing the method is shown in fig 2.2

2.5 METHODS FOR MINIMIZATION OF SUM OF SQUARES

These are special methods for minimization of functions having form of a sum of squares. Function of this type arises in a number of applications such as solving a system of nonlinear equations, curves fitting or regrssion analysis. Brief discussion of the method is given below.

The objective funtion defined as

$$M(X) = [f(X)]^T\ f(X)$$

where $f_i(X)$, $i=1,2,\ldots\ldots.m$ are nonlinear functions of X, it is assumed that $m > n$ and problem of minimizing,

$M(X)=[f(X)]^T\ f(X)$ is reduced to a linear problem by expanding $f(X)$ in a Taylor series about the current iteration point X_k and ingnoring the nonlinear terms

$$f(X_k + \Delta X_k) \approx f(X_k) + J(X_k)\Delta X_k$$

where $J(X_k)$ is the Jacobian matrix with elements

$$\frac{\partial f_i(X_k)}{\partial X_j}$$

The normal equations corresponding to this linearized problem are

$$[J(X_k)]^T\ [J(X_k)]\Delta X_k = -[J(X_k)]^T\ f(X_k)$$

and yield the connection vector ΔX_k

Iteration steps are given below:

(1) Compute $J(X_k)$

(2) Form $[J(X_k)]^T\ J(X_k) = T(X_k)$

(3) Compute ΔX_k from $T(X^k)\Delta X_k = -[J(X_k)]^T\ f(X_k]$

(4) Set $X_{k+1} = X_k + \Delta X_k$, $k = k + 1$ and repeat from step 1 till convergence is achieved. This method is called Gauss metho

As $[T(X_k)]^{-1}$ is positive definite, algorithm is potentially divergent. A new method called Descent have been proposed to overcome the difficulties of Gauss's method.

The method is decent - that is $[\Delta X_k]^T\ \nabla f(X_k) < 0$ since $\nabla M(X_k) = 2[J(X_k)]^T\ f(X_k)$

so that $[\Delta X_k]^T \ \nabla M(X_k) = 2[f(X_k)]^T \ J(X_k) \ [T(X_k)]^{-1}$

$$= [J(X_k)]^T \ f(X_k)$$

and $[\Delta(X_k)]^T \ \nabla M(X_k) < 0$

The modification requires a search along the correction vector ΔX_k to find X_{min} satisfying the min $M(X_k + \Delta X_k)$. Now new approximation to the solution is $(X_{k+1}) = X_k + \lambda min \ \ X_k$.

This modification of Gauss method is undoubtley better than the unmodified method. Still it has a serious weakness. Numerical experiments performed with the Gauss method indicated that director $\Delta(X_k)$ gives in step 3 frequently makes largest angle (nearly 90°) with the steepest direction $-\nabla M = -2[J(X_k)]^T \ f(X_k)$ This result in a high oscillatory path of iterations and very slow convergence. Levelberg (1944) and Marquardt (1963) attempted to ovecome this difficulty by introducing a new parameter $\lambda > 0$.

2.6 DEFINITION OF OPTIMIZATION PROBLEM

A general statement of the optimization prblems is as follows.

objective : Minimise $f(X)$ $X = \begin{bmatrix} x1 \\ x2 \\ . \\ . \\ xn \end{bmatrix}$

subject to

equality constraints :

$$g_j \ (X) = 0 \quad j=1,2,.....mec$$

inequality constraints

$$g_j(X) \geq 0 \quad j = mec+1, \ .....m$$

Problem of this type is called nonlinear programming problem.

CHAPTER 3

METHODS STUDIED

CHAPTER 3

METHODS STUDIED

Keeping in view the wide applicability and more advantages of methods two constrained nonlinear algorithms and one nonlinear least square method have been studied in detailed.

Methods are : Complex method , Interior penalty function method and Marquardt method.

3.1. COMPLEX METHOD

The complex method first introduced by Box (1965) is similar to the simplex method (in unconstrained nonlinear minimization), is based on heuristics search and can handle the constrained problems. The method searches for the minimum value of the objective function f(x) in a feasible region defined by lower and upper bounds (explicit constraints) $b_i \leq x_i \leq u_i$, $i=1,2,\ldots\ldots n$ and constraints function $g_i(x) \geq o$ $i=1,2,\ldots.m$ (implicit constraints).

ITERATION STEPS

(1) This method uses a set of $k \geq n+1$ points of which one point is supplied as a starting point satisfying all the constraints. the remaining k-1 points are obtained by the use of random numbers R_i in the relation

$$X_i = b_i + R_i (u_i - b_i)$$

R_i is distributed over interval (0,1). These points satisfy the lower and upper bounds constraints.

If some implicit constraints are violated then trial point is moved half way toward the centroid of the remaining already accepted points. The centroid $\bar{X}_o$ is given by

$$\bar{X}_o = \frac{1}{s} \sum_{i=1}^{s} X_i \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.1)$$

where X_1, X_2 .... X_s are available feasible points. Shifting toward centroid is repeated till a feasible point is obtained and by repeating this procedure the requisite number of k-1 points in the feasible region can be generated.

(2) The objective function f(x) is evaluated at each vertex and the vertex X_h is defined as , at which function $f(X_h)$ assumes the largest value out of the k values of f(x). Reflection of X_h is taken by computing X_r using following equation.

$$X_r = (1+\alpha\ )\ X_o - \alpha\ X_h,\ \alpha \geq 1 \qquad \ldots\ldots\ldots\ldots\ldots\ldots(3.2)$$

where X_o is centroid of remaining vertices and given as

$$X_o = \frac{1}{k-1} \sum_{\substack{j=1 \\ j\pm h}}^{k} X_j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.3)$$

Box found that $\alpha = 1.3$ was a good choice of the overreflection coefficient, but this choice is not critical. The use of over reflection coefficient $\alpha>1$, compensates for the tendency of the complex to shrink, which is caused by moves toward the centroid. Check the point for feasibility.Evaluate the function at reflected point X_r .

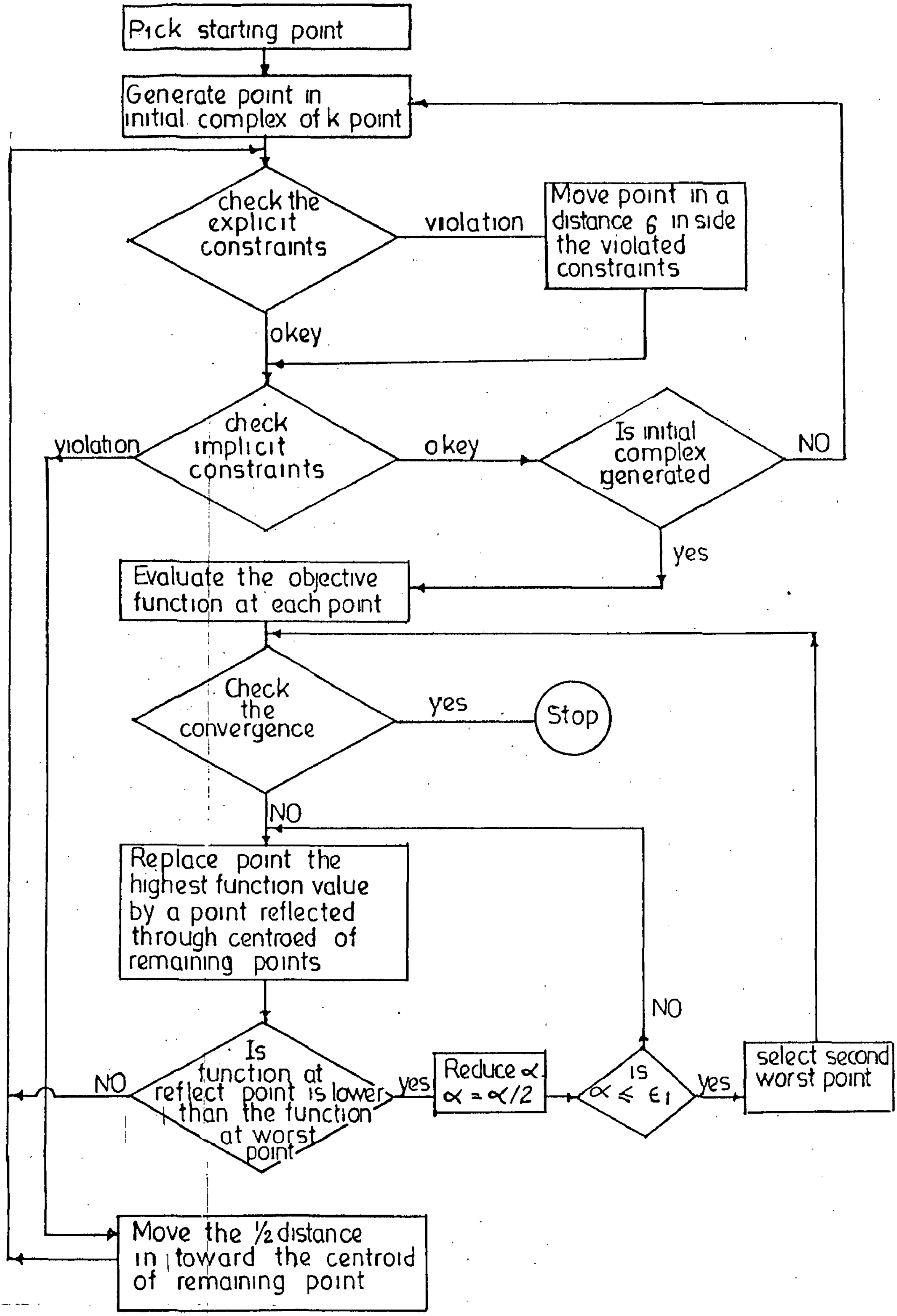

Pick starting point
Generate point in initial complex of k point
check the explicit constraints
violation
Move point in a distance б in side the violated constraints
okey
check implicit constraints
violation
okey
Is initial complex generated
NO
yes
Evaluate the objective function at each point
Check the convergence
yes
Stop
NO
Replace point the highest function value by a point reflected through centroed of remaining points
NO
Is function at reflect point is lower than the function at worst point
yes
Reduce ∝ ∝ = ∝/2
NO
is ∝ ≤ ϵ1
yes
select second worst point
Move the ½ distance in toward the centroid of remaining point

Fig.3·1. Flow chart for complex algorithm

(3) If $f(X_r) < f(X_h)$ and X_r is feasible, the point X_r replaces X_h and then return to step 2. If $f(X_r) \geq f(X_h)$, the overreflection coefficient is reduced $\alpha: = \alpha/2$, and the new x_r is computed and tried. This is repeated (if necessary) untill $\alpha \leq t$, where $t=10^{-5}$ is a satisfactory value. If the relation $f(X_r) < f(X_h)$ does not hold even for that small value of α, then the projected point X_r is replaced by original value of X_h and the second worst vertex is reflected instead. This process keeps the complex moving toward the minimum unless the centroid is very close to it. If a reflected vertex is not feasible, then this point is moved half way toward the centroid untill it becomes feasible. The method will progress so long as the complex has not collapsed into its centroid.

TERMINATION CRITERIA AND CONVERGENCE

The process is terminated when the complex shrinks to an acceptable small size or if :-

$$\left[\frac{1}{k} \sum_{j=1}^{k} [f(X_o) - f(X_j)]^2 \right]^{1/2} < \epsilon \quad ....(3.4)$$

$$\text{Where} \quad X_o = \frac{1}{k} \sum_{j=1}^{k} X_j$$

where $\epsilon > 0$ is a predetermind small number. It is important for convergence that the complex retain its full dimensionality. Box successfully applied $k = 2n$ as the number of vertices of the complex for the problems having $n \leq 5$.

DISCUSSION

The complex method is to some extent scale independent, since in initial, complex is roughly scaled to the order of the problem variables. No use is made of the first derivatives of f(X) or $g_j(X)$, hence there is relatively little work per step. The computer code is very simple and required only a moderate amount of the storage.Flow diagram for method is shown in fig.3.1

The method becomes inefficient as the number of variables increase. Another disadvantage is that it is incapable of solving problems with equality constraints.

3.2 INTERIOR PENALTY TUNCTION METHOD

Penalty methods solve the nonlinear programming problems by solving a sequence of unconstrained problems by incorporating the constraints.

Let us consider a general nonlinear programming problem.

minimize f(X) ....(3.5)

subject to

$$g_j\ (X) \leq 0, \quad j = 1,\ 2,\ \ldots\ldots.m \quad (3.6)$$

This problem is converted in to an unconstrained minimization problem by constructing a function of the form

$$\phi_k = \phi(X, r_k) = f(X) + r_k \sum_{j=1}^{m} G_j\ [g_j(X)] \ \ldots\ldots \ (3.7)$$

where G_j is some function of the constraints g_j, and r_k is a positive constant known as penalty parameter. The second term on the right side of the equation (3.7) is called penalty term. If the unconstrained minimization of the ϕ function is repeated for a sequence of values of penalty parameter r_k(k=1, 2, ...), the solution may be brouht to converge to that of the original problem stated in equations (3.5 and 3.6) for $r_k = 0$.

The penalty function formulation for inequality constrained problems can be divided into two categories, namely the interior method and exterior method. In the interior penalty method some of popularly used forms of G_j are given by

$$G_j = - \frac{1}{g_j(X)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.8)$$

$$G_j = \log [-g_j(X)] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.9)$$

The penalty term is choosen such that its value will be small at the point away from the cosntraints boundries and will tend to infinity as the constraint boundries are approached. Hence the value of ϕ function also 'blows up' as the constraint boundries are approached. Thus once the unconstrained minimization of $\phi(X, r_k)$ is started from any feasible point X, the subsequant points generated will always lie within feasible region since constraint boundries act as barrier during minimization process. This is the reason why the interior penalty function method is also known as barrier method.

In interior method, the unconstrainted minima of ϕ_k converge to the solution of equation (3.5) as r_k is varied in a particular manner.

The ϕ- function defined originally by Cassol (1961) is given as,

$$\phi(X, r_k) = f(X) - r_k \sum_{j=1}^{m} \frac{1}{g_j(X)} \quad \ldots\ldots\ldots\ldots \quad (3.10)$$

It can be seen that the value of the function ϕ will always be greater than f(x) since $g_j(X)$ is negative for all feasible points X. If any constraints $g_j(X)$ is satisfied critically the vlaue of ϕ tends to infinity. It is to be noted that the penalty term in equation (3.10) is not defined if X is not feasible.

This introduces serious short comings while using equation (3.10). Since this equation does not allow any constraints to be violated, it requires a feasible starting point for the search toward the optimum point. To solve unconstrained problem variable metric method (Davidon - Fletcher - Powell) is used and to determine the optimum step length in current search direction cubic interpolation method (one dimensional method) is used.

Iteration procedure of this method is as follows.

ITERATION STEPS

(1) Start with an initial feasible point X_1 satisfying the constraints and strict in equality sign that is $g_j(X_1) < 0$ for $j=1, 2, ...m$ and an initial value of $r_1 > 0$ set $k = 1$.

(2) Minimize $\phi(X, r_k)$ by using above mentioned unconstrained and one dimensional minimization methods and obtain the solution X^*_k.

(3) Test whether X^*_k is the optimum solution of the original problem. If X^*_k is found to be optimum terminate the process, other wise go to next step.

(4) Find the value of the next penalty parameter r_{k+1}, as

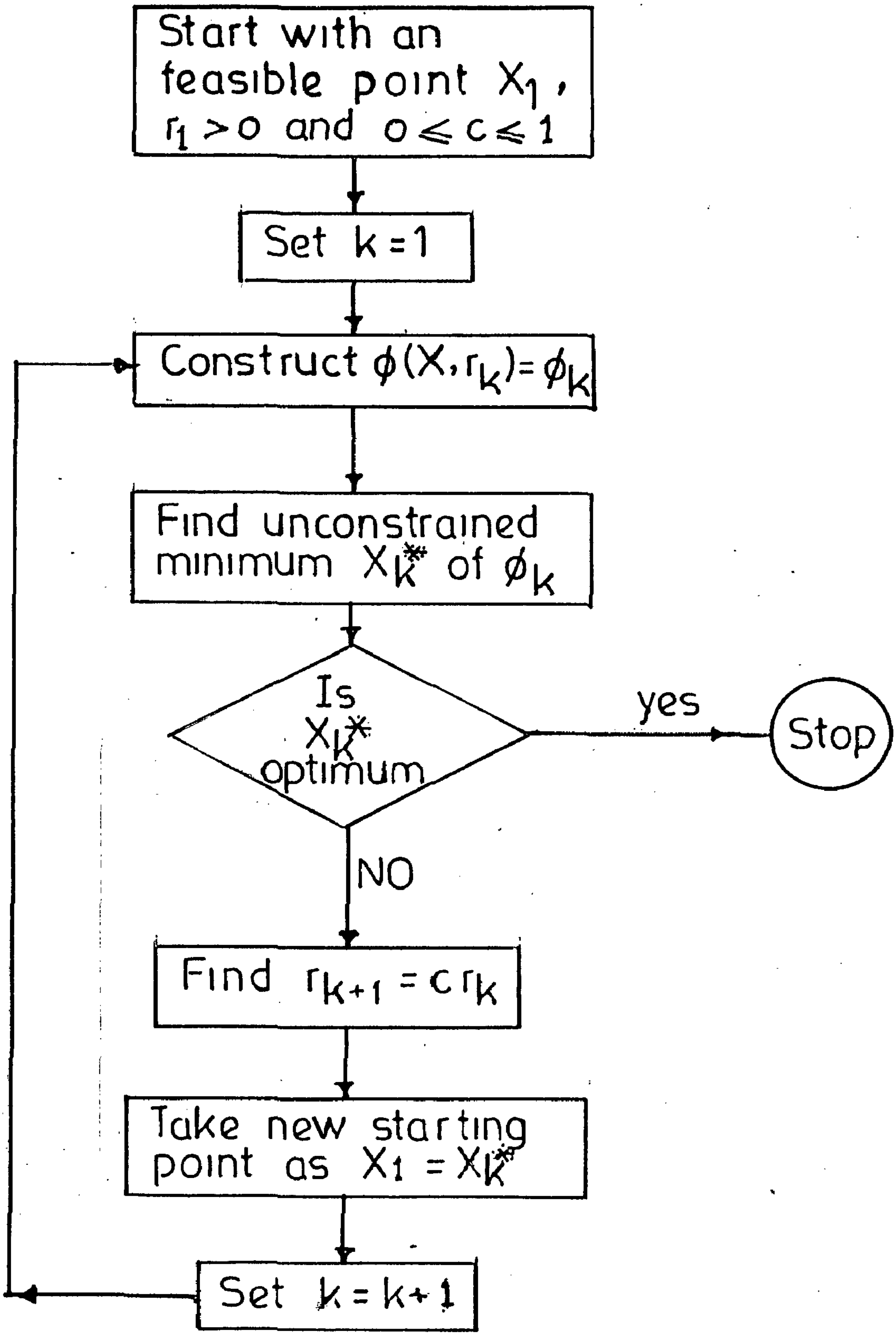

Start with an feasible point X_1, $r_1 > 0$ and $0 \leqslant c \leqslant 1$
Set $k = 1$
Construct $\phi(X, r_k) = \phi_k$
Find unconstrained minimum X_k^* of ϕ_k
Is X_k^* optimum
yes
Stop
NO
Find $r_{k+1} = c r_k$
Take new starting point as $X_1 = X_k^*$
Set $k = k+1$

Fig. 3·2. Flow chart for the interior penalty function method

$r_{k+1} = c\ r_k$ ....(3.11)

where $c < 1$

(5) Set the new value of $k = k + 1$ take the new starting as $X_1 = X^*_k$ and go to step 2

The steps are shown in the form of flow chart in figure 3.2.

The interior penalty method has been extensively studied and successfully applied to a number of practical problems. It is applicable to non convex problems and can handle nonliner constraints. The method gives feasible intermediate solutions and a final solution with desired accuracy, if a sufficiently small penalty parameter is used. The disadvantages of method are:

(1) Any special structure of constraints (e.g. linearity) is destroyed.

(2) The rapid change in the vicinity of the boundary complictes the one dimensional optimaization problem which has to be handled by special techniques.

(3) Close to boundries the term $g_j(X)$ are very large and at the final phase of minimizations, $r_k \rightarrow 0$, which causes the function $f(X,r_k)$ to be very sensitive to variable changes and introduces round off errors.

In general this method is considered suitable for solving constrained optimization problems with few highly nonlinear constraints. If, however, the objective function to be minimized subject to many linear constraints, it would be preferable to use a method that handles the linear constraints seperately.

3.3 MARQUARDT METHOD

Least square methods have number of applications such as that of solving a system of nonlinear equations, curve fitting or regression analysis. Most of the algorithms of least square estimations of nonlinear parameters use the two type of approaches. In one approach, the model is expanded as a Taylor series and improves the several parameters calculated at each iteration on the assumption of local linearity. In other approach, various modification of the method of steepest descent have been used. Marquardt (1963), Levenberg (1944). Both methods have certain disadvantages. Taylor series method because of the divergence of successive iterations and the steepest descent method because of slow convergence after the few iterations.

Marquardt (1963) proposed a new algorithms to overcome above mentioned difficulties. The method perform the optimum interpolation between the Taylor series method and gradient method.

STATEMENT OF PROBLEM

Let the model to be fitted to the data, be

$$E(y) = f(x_1, x_2, \ldots\ldots x_m; \beta_1, \beta_2, \ldots\ldots \beta_k = f(\mathbf{X},\beta) \quad (3.12)$$

where $x_1, x_2, \ldots\ldots x_m$ are independent variables.

$\beta_1, \beta_2, \ldots\ldots \beta_k$ are the values of k parameters and $E(y)$ is the expected value of the dependent variable y,

Let data point be denoted by

$(Y_i, X_{1i}, X_2, \ldots\ldots X_{mi})$

Least square function to be minimized is defined as

$$\phi = \sum_{i-1}^{n} (Y_i - \bar{Y}_i)^2 = ||Y-\bar{Y}||^2 \quad \ldots\ldots\ldots\ldots(3.13)$$

where $\bar{Y}_i$ is the value of y predicted by equation at the ith data point.

It is well known that when f is linear in β & the contours of constant ϕ are ellipsoids, when f is nonlinear the contours are distored according to severity of nonlinearity. Even with the nonlinear models, however, the contours are nearly elliptical in the immediate vicinity of the minimum of ϕ.

METHODS IN CURRENT USE

The method based upon expanding f in Taylor series is as follows

writing the Taylor series and considering only first derivative through the linear terms.

$$< Y\ (X_i\ ,\ B+\delta_t\ )\ > = f(X_i\ ,\ b) + \sum_{j=1}^{k} \left(\frac{\partial f_i}{\partial b_j}\right) (\delta_t)_j \quad ....(3.14)$$

In above equation β is replaced notationally by b, the converged value of b, being the least square estimate of β.

"The vector δ_t is a small correction to b with subscript t used to designate δ as clculated by Taylor series method". The bracket <> is used to distinguish prediction based upon the linearised model form those based upon the actual nonlinear model. Thus, the value of ϕ predicted by (3.14) is

$$<\phi> = \sum_{i=1}^{n} [Y_i - <Y_i>]^2 \quad ....(3.15)$$

Now, δ_t appears linearly in equation (3.14) and can therefore be found by the standard least square method of setting $\partial<\phi>/\partial\delta_j = 0$. j. Thus δ_t is found by solving

$$A\delta_t = g \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.16)$$

where $A_{k\times k} = P^T P$

$$P_{n\times k} = \frac{\partial f_i}{\partial b_j} , \; i= 1,\; 2,\; \ldots\ldots\ldots n, \quad j=1,\; 2,\; \ldots\ldots k$$

and

$$g_{k\times 1} = \sum_{i=1}^{n} (Y_i - f_i) \frac{\partial f_i}{\partial b_i}) = P^T (Y-f_o) \quad \ldots\ldots (3.17)$$

In practice it is found helpful to correct b by only a fraction of δ_t, otherwise extrapolations may be beyond the region where f can be adequaetly reperesented by (3.14) and would cause divergence of iterates.

The gradient methods by contrast simply step off from the current trial value in the direction of negative gradient of ϕ, thus

$$\delta_g = -(\frac{\partial\phi}{\partial b_1} , \frac{\partial\phi}{\partial b_2}, \ldots\ldots \frac{\partial\phi}{\partial b_k}) \quad \ldots\ldots\ldots\ldots\ldots\ldots (3.18)$$

Various modified steepest descent methods have been employed to compensate partially for the typically poor condition of the ϕ surface which leads to very slow convergence fo gradient method. With these gradient method as with the Taylor series method it is necessary to control the step size carefully.

CONSTRUCTION OF ALGORITHM

In Marguardt method a new parameter λ called interpolation factor is introduced. Now modified equation at r^{th} iteration becomes;

$(A^*_r + I)\ \delta^*_r = g^*_r$, ...(3.19)

Where

$$A^* = a^*_{jk} = \frac{a_{jk}}{\sqrt{a_{jj} \times a_{kk}}} \qquad \text{...............(3.20)}$$

$$g^* = g^*_j = \frac{g_j}{\sqrt{a_{jj}}} \qquad \text{....................(3.21)}$$

$$\delta^*_j = \delta_j \times \sqrt{a_{jj}} \qquad \text{.....................(3.22)}$$

where $j = 1, 2, \ldots\ldots k$

This choice of scale causes the A matrix to be transformed into matrix of simple correlation coefficients among the $\partial f_i / \partial b_j$.

This choice of scale has in fact, been widely used in linear least square problems as a device of improving the numerical aspects of the computing procedure.

This equation is then solved for δ^*_r. Then (3.22) is used to obtain δ_r. Now new trial vector given is as:

$$b_{r+1} = b_r + \delta_r \qquad \text{..............................(3.23)}$$

will lead to a new sum of squares $\phi(r+1)$, equation (3.12) and (3.13). It is essential to select λ_r such that:

$$\phi_{r+1} < \phi_r \qquad \text{..........................(3.24)}$$

It has been proved that a sufficiently large λ_r always can be found that equation (3.24) will be satisfied, unless b_r is already at a minimum of ϕ. Some form of trial and error is required to find a value of λ_r which will lead to satisfaction of (3.24) and will produce rapid convergence of the algorithm to the least square values.

At each iteration one desire to minimise the ϕ in the maximum neighborhood over which function will give adequate representation of the nonlinear functions. Accordingly, the strategy for choosing λ_r must seek to use a small value of λ_r when ever conditions are such that the unmodified Taylor series method would converge nicely.

Strategy is as follows

Let $\nu > 1$

Let λ_{r-1} denot the value of from the previous iteration

Initially Let $\lambda_0 = 10^{-2}$, say

Compute $\phi(\lambda_{r-1})$ and $\phi(\lambda_{r-1}/\nu)$

(i) If $\phi(\lambda_{r-1}/\nu) \leq \phi r$, let $\lambda_r = \lambda_{r-1}/\nu$

(ii) If $\phi(\lambda_{r-1}/\nu) > \phi r$, and $\phi(\lambda_{r-1}) \leq \phi r$, let $\lambda_r = \lambda_{r-1}$

(iii) If $\phi(\lambda_{r-1}/\nu) > \phi r$, and $\phi(\lambda_{r-1}) > \phi r$, increase by sudcessive multiplication by untill for smallest w, $\phi(\lambda_{r-1}\nu^{w}) \leq \phi r$,

let $\lambda_r = \lambda_{r-1}\nu^{w}$

By this algorithm, always a feasible neighborhood is obtained. The iteration is converged when

$$\frac{|\delta j(r)|}{t + |bj^r|} < \epsilon \quad \text{for all } j.$$

For some suitable small value $\epsilon > 0$, say 10^{-5} and some suitable t, for all j. The choice ν is arbitrary; $\nu = 10$ has been found in practice to be a good choice.

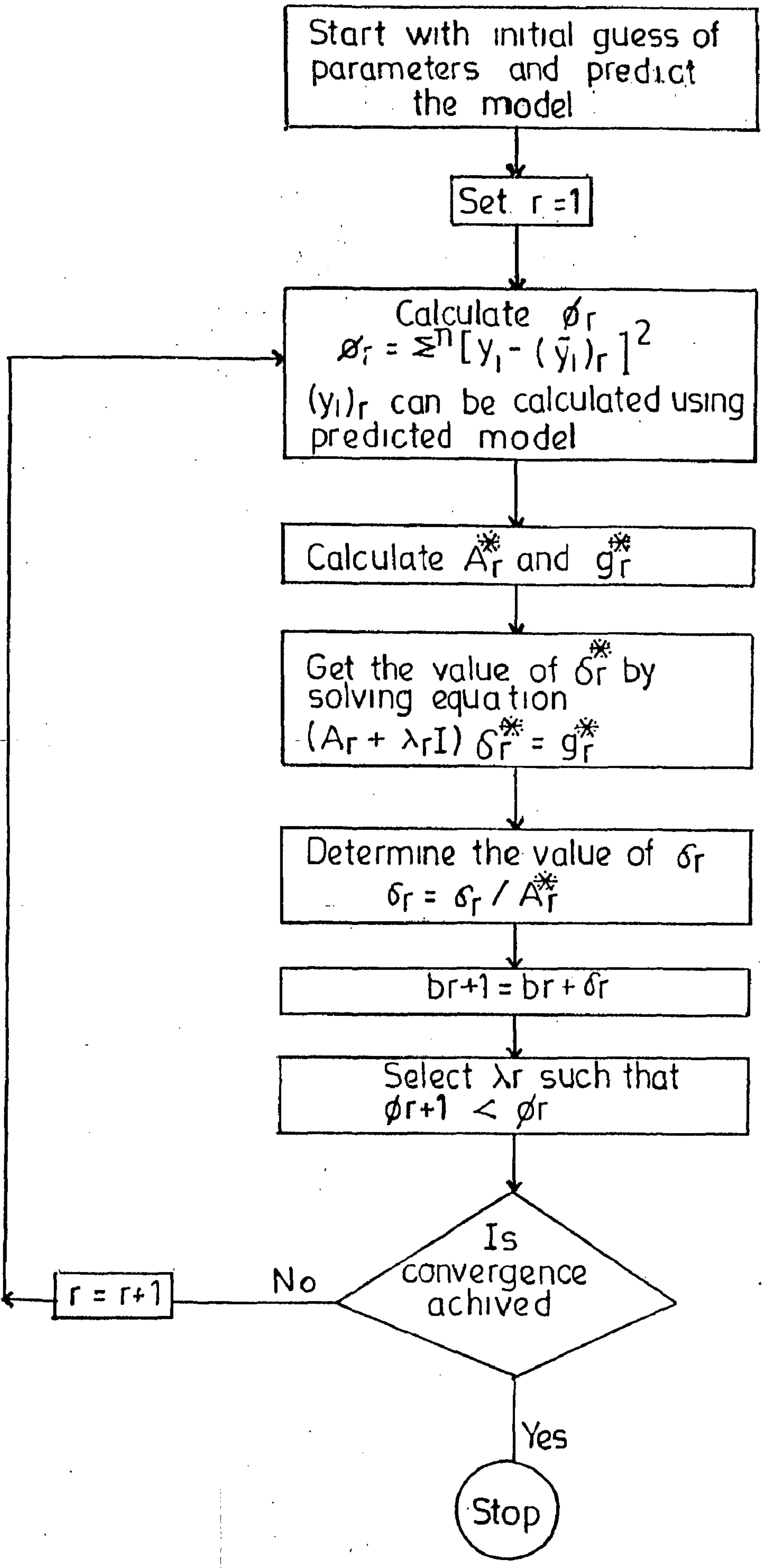

Start with initial guess of parameters and predict the model
Set r =1
Calculate ϕ_r
$\phi_r = \Sigma^n [y_i - (\bar{y}_i)_r]^2$
$(y_i)_r$ can be calculated using predicted model
Calculate A_r^* and g_r^*
Get the value of δ_r^* by solving equation
$(A_r + \lambda_r I)\, \delta_r^* = g_r^*$
Determine the value of δ_r
$\delta_r = \delta_r / A_r^*$
$b_{r+1} = b_r + \delta_r$
Select λ_r such that
$\phi_{r+1} < \phi_r$
Is convergence achived
No
$r = r+1$
Yes
Stop

Fig.3·3. Flow chart for the marquardt method

MODIFICATION IN ALGORITHM

T. Tabata and R.Lto (1976) modified Marquardt method taking into account five different values of ν. In applying the Marquardt's algorithm, the method of choosing the value of parameter λ which controls the interplation of algorithm between steepest descent method and Taylor series method has a considerable effect on the rate of convergence for some classes of problem. Rosen (1960), Ono (1971)

According to Marquardt's method, λ varies from one iteration to next by multiplying or dividing the previous by the constant factor 10, that is, the choice is made with a logarithmically constant step. To change the size of the step, the following five factors are provided in place of single factor 10 : $\nu_1 = 1.33$, $\nu_2=1.78$, $\nu_3=3.16$, $\nu_4= 10$ and $\nu_5=100$. One of these factors ν_i is selected for each iteration defiending upon the history of minimization. The history is defined as a sequence of results of comparing the sum of the squared residuals ϕ with its least value so far obtained and consists of the latest three results except at the earliest iterations. The scheme used in present study to determine the value of subscript i of the ν-factor is givin in Table 3.1.

TABLE 3.1

Scheme for determining the value of i of theν-factor from previous value io (D and I) stand for the decrease and increase respectively, of ϕ, and DI, for example means that ϕ was decreased and then increased at the latest two iterations.

Conditions		
History	i_o	i
At the start	--	4
DI, ID	--	i_o-1
DDI, IDI, IID	$i_o>1$	i_o-1
DDD	$i_o<5$	i_o+1
III	$i_o<3$	3
All the other cases		i_o

The algorithm described shares with the gradient methods, their ability to converge from an initial guess which may be outside the region of convergence of the other methods. The algorithm shares with the Taylor series method the ability to close in on the converged values rapidly after vicinity of the converged values has been reached. Thus the method combines the best features of previous methods while avoiding their most serious limitations. Flow chart for Marquardt method is shown in fig. 3.3

The listing of computer programs are available in Chemical engineering department of University of Roorkee,Roorkee. The features of the computer programs are given in Appendix -A

CHAPTER 4

OBJECTIVE FUNCTIONS STUDIED

CHAPTER 4

OBJECTIVE FUNCTIONS STUDIED

To check the reliability and functioning of the algorithms were tested for different non-linear function. These functions were selected for the requirements in the practical area of practical engineering field.

4.1 TESTED OBJECTIVE FUCTIONS

As the algorithm code were not developed but were adopted from the standard text book (Kuester and Mize (1973) and Rao (1987)) These fortran codes wrer initially tested for the known function with the given solution. These function are

(i) Minimixe $f(x) = x_1^2 - 6x_1^2 + 11x_1 + x_3$ .....(4.1)(i)

subject to

$x_1^2 + x_2^2 - x_3^2$

$4 - x_2^2 - x_2^2 - x_3^2 \leq 0$

$x_3 - f \leq 0$

$- x_i \leq 0 \quad i = 1, 2, 3$

This minimization problem was tested using complex method and interior penalty function method.

(ii) Parameter A_1, A_2 and A_3 were estimated for the model

$$\hat{Y} = A_1 + A_2 \; e\hat{x}p(A_3 X) \text{...........} (4.1)(ii)$$

for the known $\hat{Y}$ and $\hat{X}$ data points. Thus data points are given in following table.

$\hat{Y}$	157	151	379	421	460	426
$\hat{X}$	-5	-3	-1	1	3	5

This problem was used to test the Marquardt method while utilizing non-linear least square method.

Objective functions based on following area have been selected for detailed study

1 Parameter estimation.

2 Sequential experimental design.

4.2 PARAMETER ESTIMATION

An important problem in chemical reaction engineering is the estimation of the parameters of a kinetic model from experimental data.

In the mathematical model, there are dependent and independent variables and also certain constants. The constants are generally called parameters.

In experiments the dependent variables may be measured diretly but the parameters can not be. Approximate values of the parameters are calculated from dependent and independent variables. Since only approximate parameters values are found. The parameter are said to estimated.

4.2.1 FORMULATION OF FUNCTION

The ethylene hydrogenation reaction system studied by Barton (1976) was used for the present study. The rate model for reaction was

$$r_{C_2H_6} = Ao \exp(-E/RT) (X_{C_2H_4})^{\alpha} (X_{H_2})^{\beta} \quad ....(4.2)$$

Where $r_{C_2H_6}$ is the rate of formation of ethane and $x_{C_2H_4}$ and x_{H_2} are mole fractions of ethylene and hydrogen respectilvely.

The original parameter estimates refered to as 'true' values were taken from Barton's analysis and are shown in table 4.1.

Table 4.1 True values of Parameters

Parameter	Value
Ao	4.629×10^3
ϵ	11.6 k cal / mole
α	0.33
β	1.0

The data from each set of experiments consist of one set of independent variabls (mole fraction of ethylene and hydrogen and reaction temp.) together with the dependent variables obtained by adding errors to the true values calculated from equation (4.2).

The errors used were calculated from

% Error = random number x noise level

The random numbers were generated from a Gaissian normal distribution with zero mean and a constant standard deviation of 0.5 using following formula.

$$f = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[- \frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2 \right]$$

μ = mean

σ = standard deviation

10% and 30% noise level were used to test the performance of selected algorithms

The objective function for the parameter estimation is a non - linear least square function and is defined as follows.

$$\text{Minimise } U_{estimation} = \sum_{u=1}^{n} (Y_u - \bar{Y}_u)^2$$

Where Yu is observed rate and is calculated as follows.

$Y_u = Y_{true} - Y_{true} x error/100$

Y_{true} was calcuted using true values of parameters for different experimental data points and $\bar{Y}_u$ is the expected value of rate given by Equation 4.2

4.3 SEQUENTIAL EXPERIMENTAL DESIGN

Parameter estimation for a model of known form has been discussed above. The estimates thus obtained are likely to be approximate, since experiments are probably poorly designed over the desired experimental range. Box and Lucas (1959) developed an experimental design procedure for decreasing the amount of uncertainity associated with the estimates of parameters. It aims at reducing the volume of the joint confidence region (JCR) with the extimates.

Box and Hunter (1965) established a criterion for a sequential experimental design, generally known as minimum volume design critaion (M V D), where all the available experimental results are analysed each time as an additional experiment is performed and the current information is then used for design of the next experiment. Some applications of this criterion to real experimental studies in the field of chemical kinetics have been reported by Mezaki (1969), Froment and Mezaki (1970) and Graham and Stevenson (1972).

4.3.1 DESIGN CRETERION FOR SEQUENTIAL EXPERIMENTAL DESIGN

Let us consider a rate model given by following equations

$$Y_u = f(x_u, k) + \epsilon_u \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.3)$$

where Y_u is the measured reaction rate,

$X_u = (x_{1u}, x_{2u}, x_{3u}, \ldots\ldots x_{mu})$ are m operating variables for experimental run u, u =1, 2, ...n and $K = (k_1, K_2, \ldots. k_p)^T$ are p parameters. If the random errors ϵ_u, u=1, 2, ...n are indepently and normally distributed with constant variance σ^2, then the variance - covariance matrix of the least square estimates

$\bar{K}$ is

$$V = (D^T D)^{-1} \sigma^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.4)$$

where D is an (nxp) matrix. An element of D d_{uj} is the partial derivative with respect to the j^{th} parameter evaluated for u^{th} experimental condition of X_u at the least square estimates $K = \bar{K}$

$$du_j = \left[\frac{\partial f(x_u,k)}{\partial k_j}\right] K = \bar{K} \text{ .... } (4.5)$$

The MVD creterion for optimal experimental design i.e.

Minimise

$$U_{design} = 1/|(D^T D)| \text{ .... } (4.6)$$

For the given model (Equation 4.2) the elements of the D matrix are definedas follows.

$$\partial(r_{C_2H_6})/\partial A_0 = \exp(-E/RT)\ (X_{C_2H_4})^{\alpha}\ (X_{H_2})^{\beta} \text{ ....}(4.7)$$

$$\partial(r_{C_2H_6})/\partial E = -r_{C_2H_6}/RT \text{.....................}(4.8)$$

$$\partial(r_{C_2H_6})/\partial \alpha = r_{C_2H_6}\ \ln(X_{C_2H_4}) \text{............}(4.9)$$

$$\partial(r_{C_2H_6})/\partial \beta = r_{C_2H_6}\ \ln(X_{H_2}) \text{..............}(4.10)$$

The initial sets of independen variable to design the 7th, 15th and 25th experiment were used from the simulated studies of Agarwal and Brishk (1985). The continuous operation region was choosen on the independent variable and is defined as fellow :

$$0.1 \le X_{C_2H_4} \le 0.4 \text{.................}(4.11)$$

$$0.4 \le X_{H_2} \le 0.8 \text{.................}(4.12)$$

$$313 \le T \le 363 \text{................}(4.13)$$

Constraints from the actual experiments were used to bound the allowable space. Thus an upper limit of the reaction rate was set at 1.5 x 10^{-4} kg /(kg catalyst)-s. Because higher reaction rate gave run away condi tion (Barton 1970). The mole fractions were restricted to $X_{C_2H_4} + X_{H_2} < 1.0$ with nitrogen used as inert diluent when necessary

CHAPTER 5

RESULTS AND DISCUSSION

CHAPTER 5

RESULTS AND DISCUSSION

This chapter discribes the result obtained from testing the three selected algorithms (complex, Interior penalty function and Marquardt method) discusses their performance and compares the effectiveness for two different objective functions.

The computer codes of the selected algorithms have been adopted from the standard books (Kuester and Mize (1971) and (Rao (1987)). Computer programs were run on Unicomp PC- AT.

Before implementing the actual objective function, selected algorithms were tested for the test problems with known solution to ensure their performance in solving the optimization problems. Tables 5.1 and 5.2 show the performance of selected algorithms.

Table 5.1 Solution of Tested problem 4.1 (i)

Methods used	Starting function value	Final function value	number of function evaluations	Actual solution
Complex method	3.041	1.435	125	1.414
Interior penalty fuction method	3.041	1.426	2448	

Table 5.2 Solution of Tested problem 4.1 (ii)

Methods used	Starting function value	Final function value	number of function evaluations	Actual solution
Marquardt method	7.546E6	1.339E6	81	1.339E6

It is evident from the Tables 5.1 and 5.2 that the optimum solution are close to the known solution. This showed that all computer codes adopted were working perfectly alright.

The different algorithms were, then applied to the following two objective functions in area of

(1) Parameter estimation.

(2) Sequential experimental design.

5.1 PARAMETER ESTIMATION

Complex and Marquardt methods were tested for the estimation of kinetic parameters for the model described by equation 4.2 Three sets of data points were taken in order to cover the range of the situation.6, 14 and 24 data points were used. These data points are shown in Table B-1 of Appendix B.

The simulated rate data points were generated based on two noise level, 10% and 30%. The 10% noise level covers the maximum possible range in a laboratory experimental situation while, 30% noise level approximates the industrial situation.

Table 5.3 and 5.4 compare the result obtained for 10% and 30% noise level.

Table 5.3 **Comparison of Marquardt and Complex method for 10% noise level**

Methods used	Starting function value	Optimum function value	Number of data points	Function evaluations	Ratio of optimum to starting valu
Marquardt method	1.023E-8	2.6912E-12	6	87	2.63E-4
	3.887E-8	1.579E-11	14	76	4.06E-3
	9.813E-8	1.074E-10	24	66	1.09E-3
Complex method	1.023E-8	2.423E-11	6	92	2.37E-3
	3.887E-8	4.960E-10	14	86	1.28E-2
	9.813E-8	1.096E-10	24	86	1.12E-2

Table 5.4 **Comparison of Marquardt and Complex method for 30% noise level**

Methods used	Starting function value	Optimum function value	Number of data points	Function evaluations	Ratio of optimum to starting valu
Marquardt method	7.516E-9	2.730E-11	6	79	3.63E-3
	2.796E-8	2.487E-10	14	48	8.89E-3
	6.925E-8	9.663E-10	24	47	1.4E-2
Complex method	7.516E-9	5.193E-11	6	78	6.91E-3
	2.796E-8	5.036E-10	14	101	1.8E-2
	6.925E-8	3.028E-9	24	133	4.37E-2

It is evident from the Tables 5.3 and 5.4 that Marquardt method posseses the superiority over the complex method in minimizing the sum of squares. As it is also evident from the Tables 5.3 and 5.4 that the complex method could not achieve the minima obtained by Marquardt method.

The ratio, optimum value to starting function value a measure of reduction in the function value, Tables 5.3 and 5.4 show these values.

As expected 30% noise level showed a smaller ratio values as compared to 10% noise level (Table 5.3 and 5.4). At the same time it is also evident from Tables 5.3 and 5.4 that the Marquardt method is superior to the complex method in both cases 10% and 30% noise level.

Complex method searches the minima in the direction which is reflection of worst point in the feasible region so there could be a possibility that complex method could collapse in to subspace and unable to find the solution in the desirable space of variables. Also it could shrink drastically in a steep valley and terminate at undesirable optimum.

Fig. 5.1 to 5.6 show the behaviour of algorithms. The values were plotted between the function value and number of function evaluations for different noise level. It is evident from these figures that both Marquardt and Complex method reached near to optimum value rapidily in few number of function evaluations. As they led in vicinity of the optimum , convergence became rather slow and took larger function evaluations to reach the actual optimum.

5.1.1 EFFECT OF STARTING POINT SELECTION

The algorithms were also tested for different starting points to check to whether the optimizers were able to find the common optimum point. Different starting point were taken to

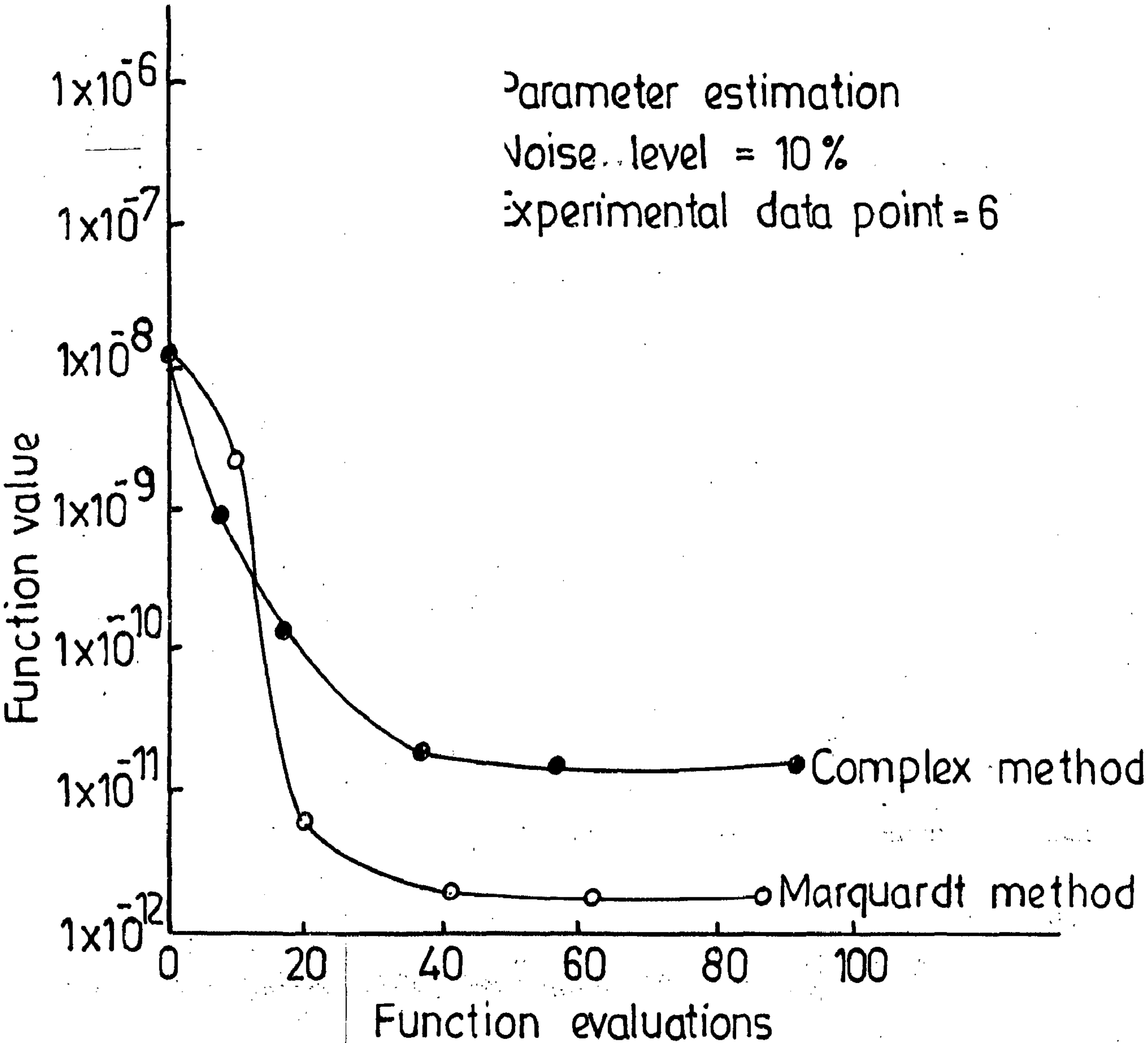


Fig.5.1. Variation of function with function evaluation

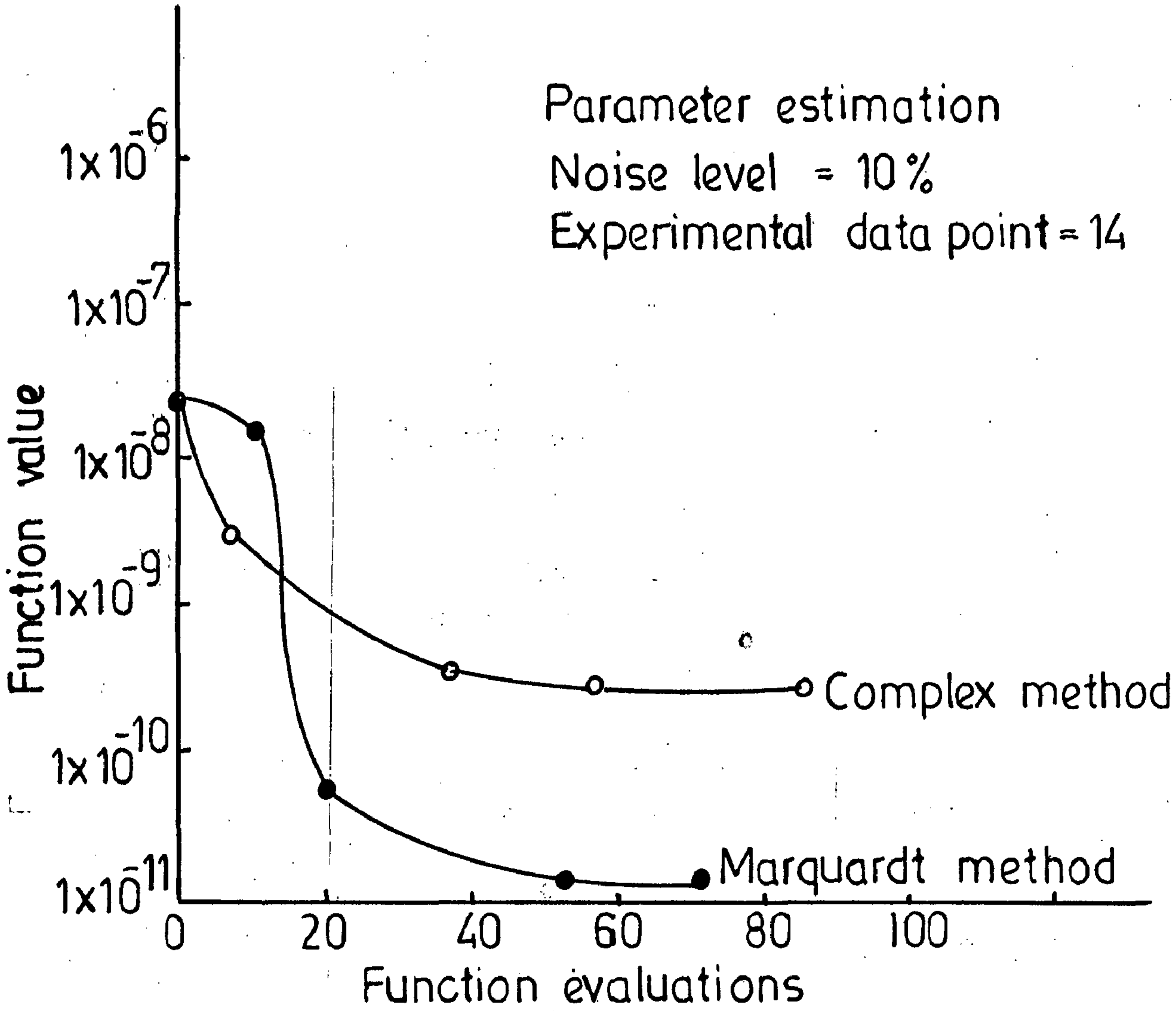


Fig. 5·2. Function variations with functior evaluation

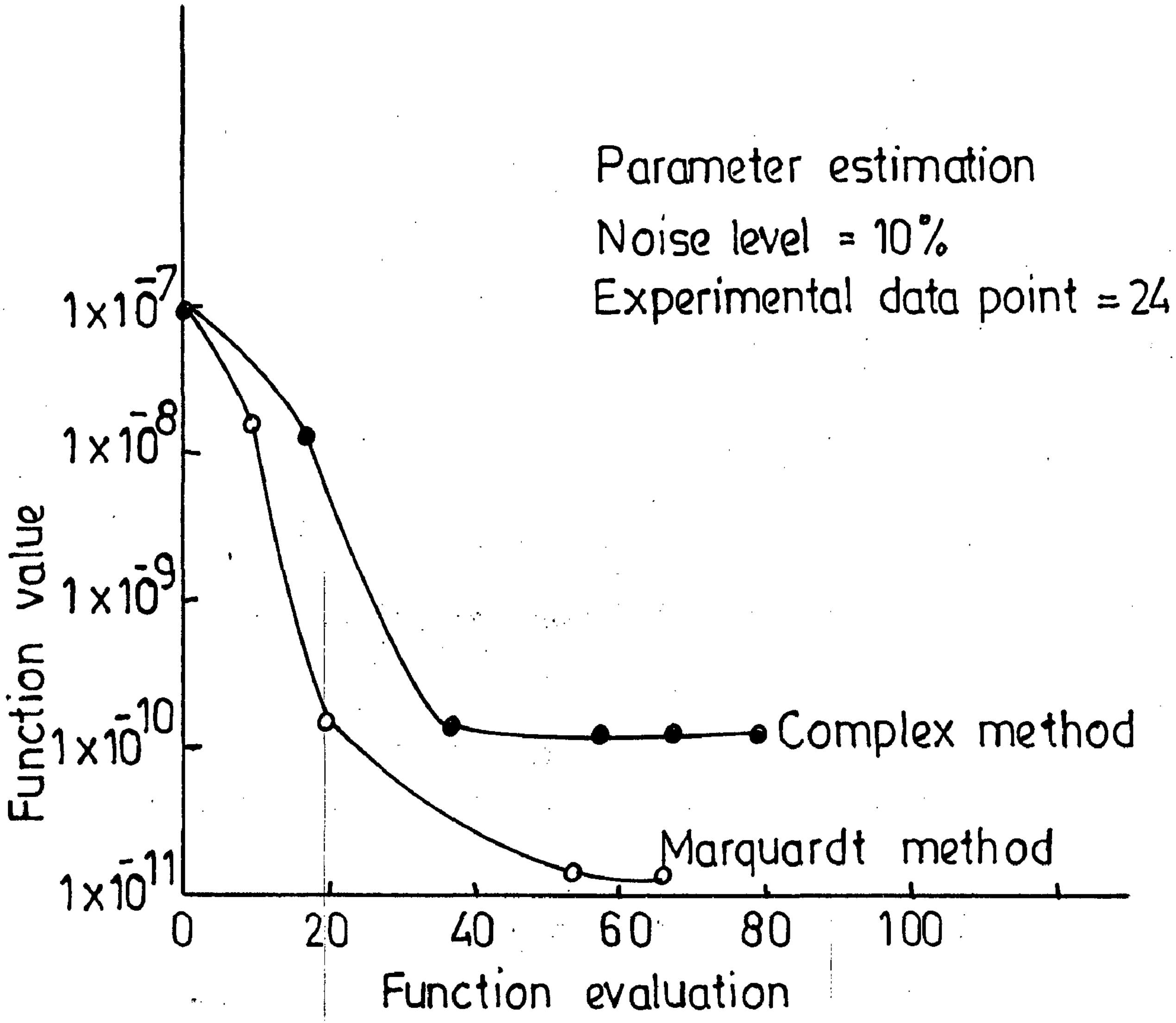


Fig.5.3 .Function variations with function evaluation

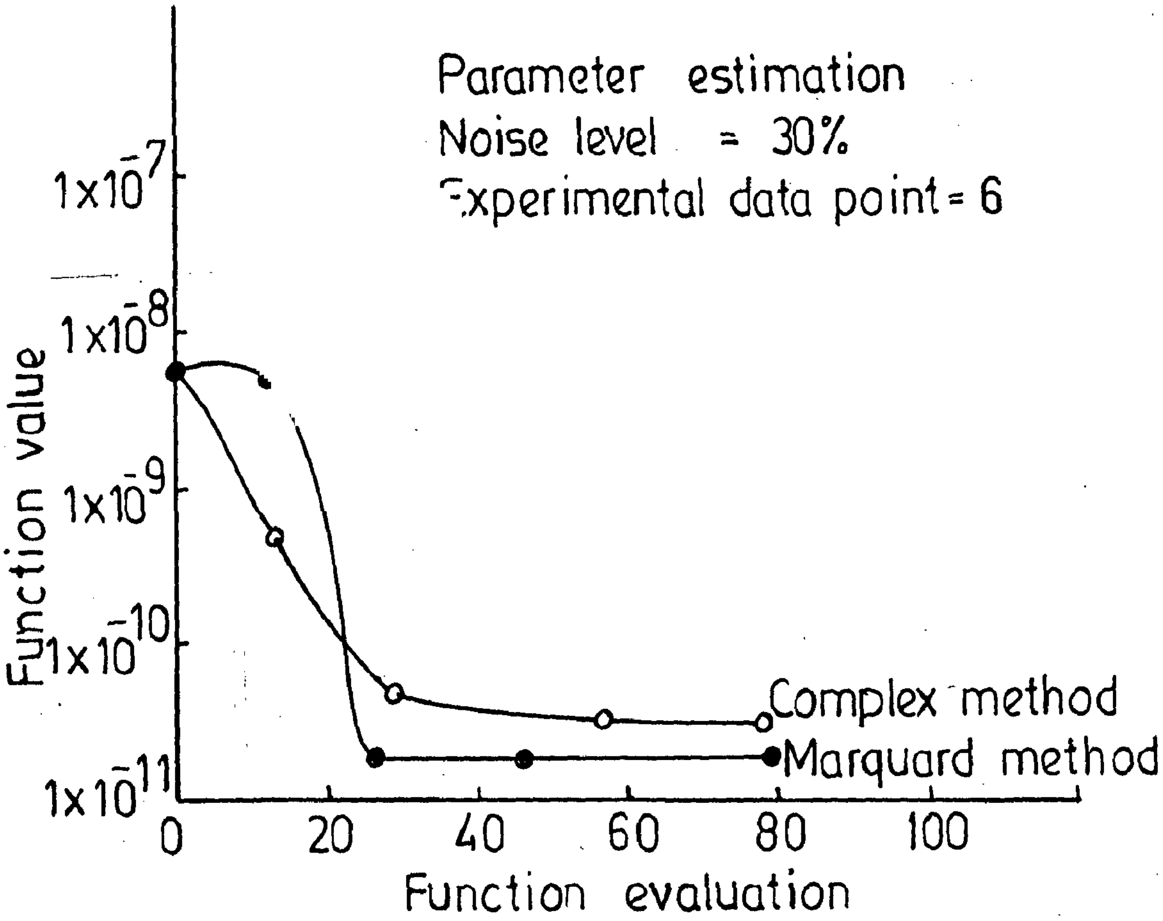


Fig.5·4. Function variation with function cvaluations

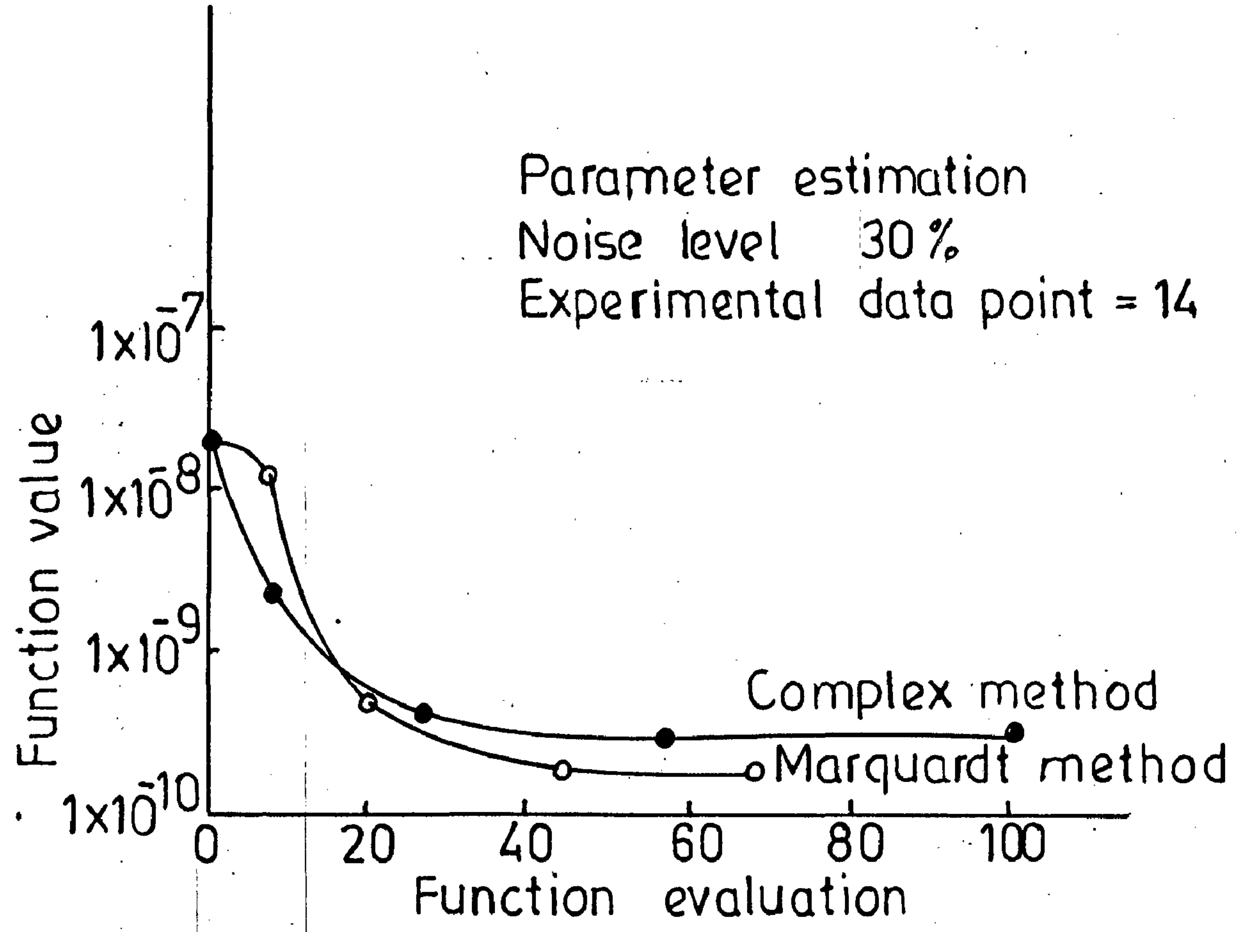


Fig. 5·5 . Function variation with function evaluation

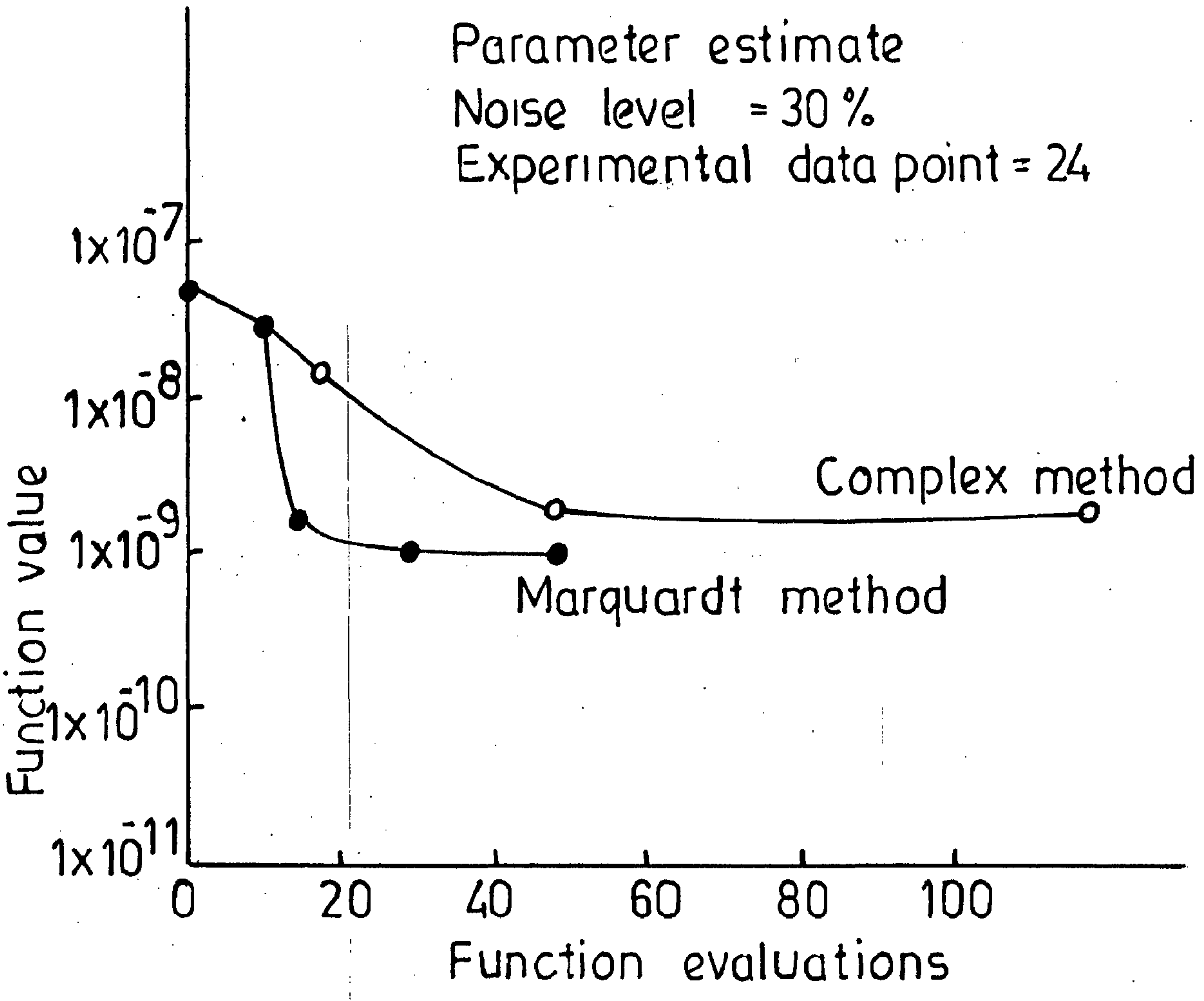


Fig. 5·6. Function variation with function evaluation

cover the range of the parameters. This behaviour was tested using 14 data points of 10% noise level. Table 5.5 shows the results.

Table 5.5: Comparison of the methods different starting points

Methods used	Starting Points (Ao, E, α, β)	Starting function value	Optimum function value	Number of function evaluations
Marquardt method	3300,9500,.1,.2	1.207E-4	1.57E-11	153
	5800,1.45E4,.95,1.8	9.908E-8	1.57E-11	123
	3900,1.0E4,.2,.5	1.782E-5	1.57E-11	123
	5000,1.3E5,.7,1.5	8.796E-8	1.57E-11	104
	4000,1.2E5,.5,1.3	4.578E-8	1.57E-11	79
Complex method	-do-	-do-	1.345E-9	65
			3.364E-9	165
			9.555E-10	98
			1.075E-9	81
			3.148E-9	63

*14 data points of 10% noise level.

It is evident from the Table 5.5 that Marquardt method reaches to same optimum point even taking the worst starting point. However, the complex fails to achieve the same.

The above result and discussion leads to the conclusion that Marquardt method is undoubtedly better choice in minimizing the sum of squares of residuals. Also Marquardt could be operated till worst starting points.

5.2 SEQUENTIAL EXPERIMENTAL DESIGN

Complex method and interior panalty function method were tested for sequential experimental design problem. As Marquardt method can not be employed for this type of optimization problem, three experiments were designed, namely 7th, 15th and 25th. Table B-1 of the Appendix B shows the different experimental points.

Table 5.6 shows the result obtained using the two optimizers.

Table 5.6: Comparison of complex and interior penalty function method

Methods used	Starting function value	Optimum function value	Experiment number designed	Function evaluations
Complex method	35.674	1.337	7	63
	0.449	0.422	15	48
	0.277	0.268	25	41
Interior penalty function method	35.674	2.398	7	1936
	0.449	0.423	15	1776
	0.277	0.269	25	1656

It is evident from the Table 5.6 that except 7th experiment both method approach to same final function value. For the 7th experiment this could be due to less number of data points and therefore movement of optimizer is confined in narrow region. However as number of experiments increased it became rather easy to find the optimum point for the optimizers. It is also evident from Table 5.6 that complex method takes lesser number of function evaluations than interior penalty function method. Complex method is a gradient free algorithm. Therefore algorithm does not calculate the derivatives of the function and evaluates the function once at any iteration. On the other hand interior penalty function method utilized the variable metric method of unconstrained minimization. In variable metric method, the first derivates are required to establish the search directions, derivatives may be determined analytically or numerically. Numerical technique was used to calculate the derivatives for the present function as analytical derivatives were not possible. Newton's forward difference was used to calculate the numerical derivatives. Newton's forward difference is given by equation

$$\frac{\Delta f}{\Delta x} = \frac{f\ (x + \Delta x) - f\ (x)}{\Delta x}$$

where Δx is increment in variables.

To calculate the derivatives using numerical technique function has to be evaluated n times (n number of variables). One dimensional method is used to determine the optimum step size in current search direction. Here cubic inter polation method was used to set the optimum step length. In cubic interpolation method two points A and B are located in such a manner that at A $f'_A < 0$ and at B, $f'_B > 0$ where $f'_A$and f'_B are derivative at A and B with respect to step length and are calculated as follows.

$$f'_A = \sum_{i=1}^{n} (\nabla f_i\ )_A\ S_i$$

Sequential experimental design
design of 7th experiment

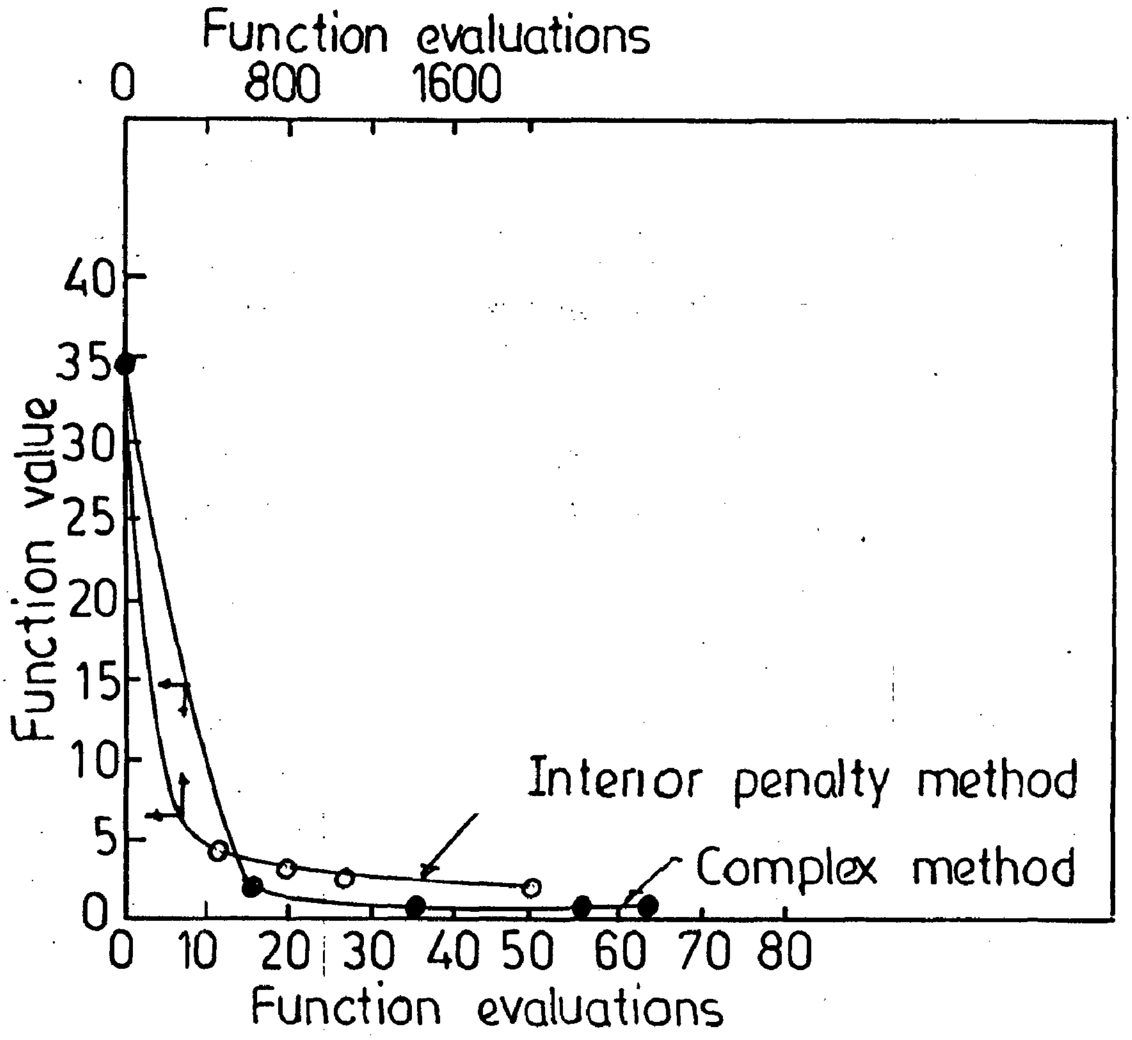


Fig. 5·7. Function variation with function evaluation

Design of 15 th experiment

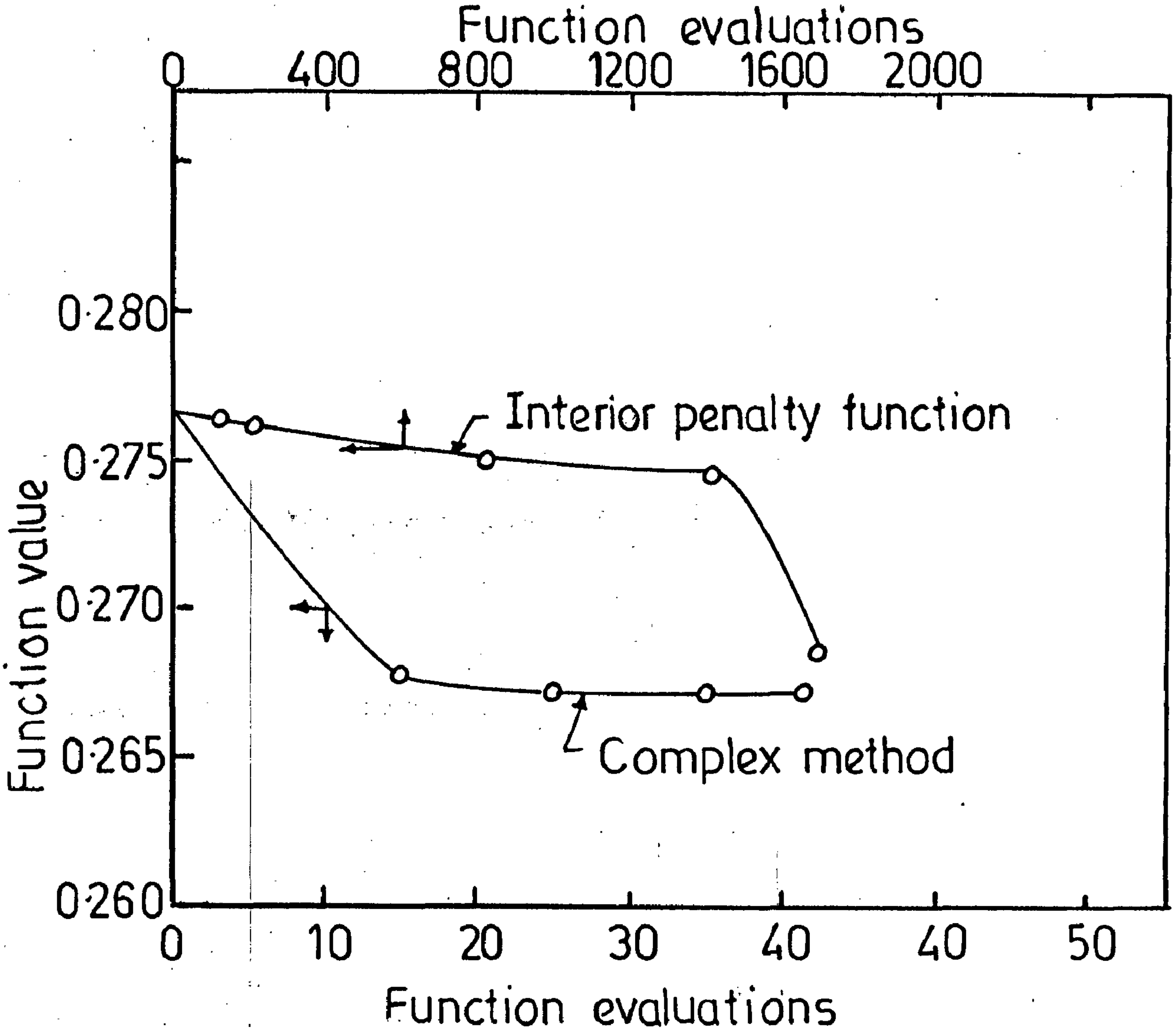


Fig. 5·8. Function variation with function evaluation

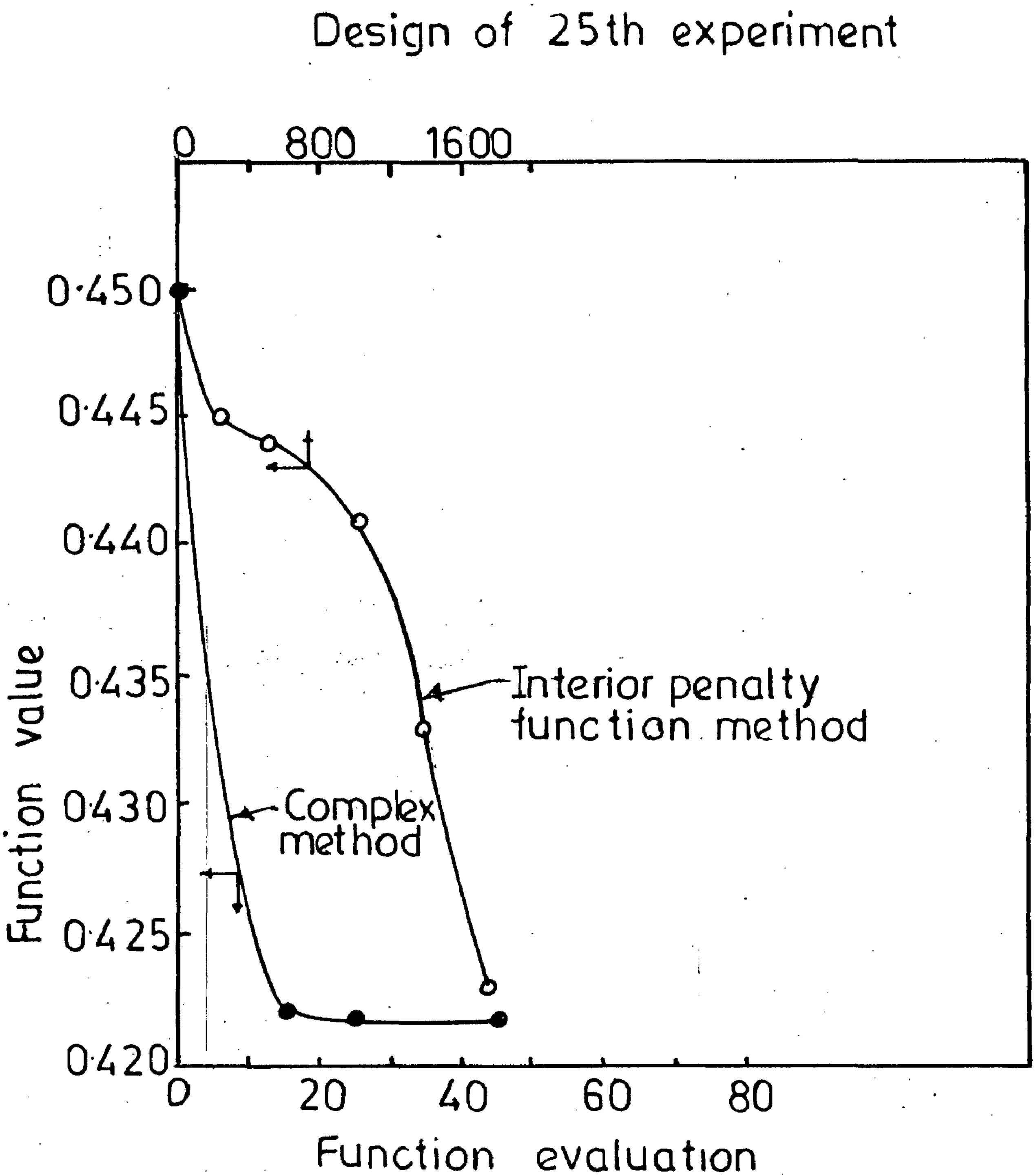


Fig. 5·9 . Function variation with function evaluation

$$f'_B = \sum_{i=1} (\nabla f_i)_B \, S_i$$

where S_i, search direction for ith component.

In locating point B, many times derivatives were calculated depending upon increase and decrease in step length. Which, inturn ,was necessitated due to increasing negative value of the derivative in the direction of descent and followed by an extremly sharp change in slope, in the proximity of the constraints. It caused the function to behave in concave fashion making it difficult to locate point B.Consequently it resulted in increasing function evaluations. It is apparent from the above discussion that interior penalty method requires the higher function evaluations in reaching to an optimum solution. On the other hand complex method takes fewer function evaluations (Table 5.6).

Fig. (5.7-5.9) shows the performance of these two algorithms. It is evident from the figures that complex method takes fewer function evaluations to reach in vicinity of the optimum solution then takes longer time to converge.

Similiar behaviour was also observed for the parameter estimation problem. Whereas except for design of 7th experiment, interior penalty function method performed slowly at the initial stage and then reduced the function drastically.

Above result and discussion for the sequential experimental design function leads to conclusion that complex method appears to be superior over interior penalty function method.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

Three algorithms of non-linear optimizer , Complex, Marquardt and Interior penalty, were used to study their performance and effectiveness for two known non-linear functions in the field of reaction engineering namely, parameter estimation and sequential experimental design.

Complex and Marquardt methods were utilized to estimate the kinetic parameters for ethylene hydrogenation reaction rate model (power-law model). Marquardt method showed superiority over Complex method. Also, Marquardt method showed that it can utilize the worst starting points in achieving the optimum solution. However, complex failed to achieve the minima obtained by Marquardt method.

Complex and Interior penalty function methods were used to design the experimental variables for the ethylene hydrogenation reaction. Both optimizers achieved the same optima but interior penalty function took large number of function evaluations.

Interior penalty function method used along with variable metric method and cubic - interpolation method requires much higher number of function evaluations, due to two primary reasons.Firstly, calculation of derivative required in choosen unconstrained multivariable and single variable method and secondly, in locating the second bound for the location of minima in one dimensional search.

Complex method for both approached in the vicinity of the optima with fewer function evaluations but took long time to converge.

6.2 RECOMMENDATIONS

Marquardt method is recommended to be utilized in the estimation of parameters. Complex method could be recommended for the sequential experimental design problems. Also,Complex could be utilized in achieving the region of minima and there after a better optimizer could be used to calculate the optimum solution.

To reduce the number of function evaluations for interior penalty function method, the unconstrained optimization technique be replaced by a gradient less technique, such as Powell (1964) method and cuic interpolation method be replaced by Golden Section method or Fibonacci method

A large variety of problems involving higher number of variables and different type of objective functions and constraints be used to compare the performance of non linear optimizers.

REFERENCES

REFERENCES

1. Agarwal, A.K.,"PhD thesis", University of Sydney, Sydney. (1984)

2. Agarwal, A.K., Brisk, L.M., "Sequential experimental Design for precise parameter estimation", Ind. Eng. process Des. & Dev., Vol. 24, pp. (203-207), (1985).

3. Betts, J.T. "An improved Penalty Function Method For Cosntrained Parameter Optimization Problems" , J. of Optimization Theory and Application , Volume 6, No. 1/2, page 1-24, (1975).

4. Box, M.J. " A comarison of several current optimization methods and the use of transformations in constrained problems", Computer Journal, vol.9, page 67-77, (1966).

5. Box, M.J. " A new method of constrained optimization and a comparison with other methods", The computer Journal, vol.8, p-42, (1965).

6. Conn, R, Andrew, "An algorithm for least squares estimation of Nonlinear parameters" J. Soc. Ind. Appl. Math, volume - 4, No. 2, (1963).

7. Davidon, W.C., "Variable metric method of minimization" ,Argonne National Laboratory Report No. ANL-5990, (1959).

8. Dixon, "Nonlinear optimization", The English University Press Limited.

9. Fiacco, A.V., McCormic, G.P.,"Nonlinear programming: Sequential unconstrained minimization technique",Willey, New York, (1968).

10. Fletcher, R. "A new approach to variable metric method", The Computer Journal, Volume 13, pp. 317-322 (1970).

11. Fletcher, R., Powell, M.J.D. "A rapidly condvergent method for minimization", The Computer Journal, volume 6, No. 2, pp. 163-168, (1963).

12. Himmelbau, D.M., "Applied Nonlinear programming", McGraw - Hill Book Company.

13. Lasdon, L.S.S., Kowalc, J.S., Pizzo, J.T.,"Iterative methods for Nonlinear optimization problems", Prentice Hall Inc.

14. Lasdon, L.S." Survey of Nonlinear programming applications", Operation Research, Volume 28, No.5, pp.1029-1039, (1980).

15. Mahoney, J.F., " Comments on sufficiency conditions for constrained extrema", AIChe Journal, volume 25, No.2, pp. 370-373, (1979).

16. Marquardt, D.W., "An algorithm for least squares estimateion of nonlinear parameters", J. Soc. Ind. Appl. Math, vol.4, No.2, (1963).

17. Meyer. G.L., "An Efficient method of feasible directions", SIAM. J. Control and optimization, volume 21, No. 1, (1983).

18. Mottl, J.," Description of a program for nonlinear programming", The Computer Journal, vol. 22, No. 3, pp. 256-261, (1977).

19. Nelder, J.A., Mead, R., "A simplex method for function minimization", The Computer Journal, vol. 4, pp. 441,(1962).

20. Osborne, M.R., Ryan, D.M., " On penalty function methods for nonlinear programming problems", vol. 31,pp. 559-578,(1970).

21. Rao, S.S., "Optimization theory andapplications", Willey Eastern Limited.

22. Rosenbrock, H.H., " An efficient method for finding the greatest or least value of a function", The Computer Journal, vol.3, pp. 175-184, (1960).

23. Umeda, T., " A modified complex method for optimization", Ind. Eng. Chem. Process Dev. & Des. ,volume 10, No. 2, pp. 229-236, (1971).

24. Tabata. T., " Effective treatment of the interpolation factor in Marquardt's nonlinear least square fit algorithm", The computer Journal, Vol. 18, No.3, pp. 250-251, (1973).

25. Zoutendijk, G., "Nonlinear programming: A numerical survey", J. SIAM. Control, vol. 4, No. 1, pp. 194 - 210, (1966).

APPENDIX A

A1. COMPLEX METHOD

A1.1 PROGRAM DESCRIPTION

A 1.1.1 USAGE

The program consists of main progoram ,four general subroutines (CONSX, CHECK, CENTR, RANDOM) and three user supplied subroutines (FUNC, BOUND, CONST), initial guesses of independent variables, solution parameters, dimension limit and printer code designation are passed to the subroutines from the main program. Final function , independent variables values and function evaluations are transfered to main program for print out. Subroutine CONSX is the primary subroutine and coordinates the special purpose subroutines CHECK, CENTR, FUNC, CONST and RANDOM. Intermediates print-outs are provided in this subroutine, if the user desires. Format changes may be required depending on the problem under consideration.

A 1.1.2 SUBROUTINE REQUIRED

SUBROUTINED CONSX (N,M,K, NMAX, I, IEV1, IT, ALPHA, BETA, GAMMA, DELTA, FF, EPS1, X, XTEMP, F, G, H, IPRINT) called from main program and coordinates all the special purpose subroutine (CHECK, CENTR, FONC, CONST, BOUND, RANDAM).

SUBROUTINE CHECK (N,M,K,X,XC,I, KODE, G,H) checks all the point against explicit and implicit constraints and applies correction if violated.

SUBROUTINE CENTR (N,M,K, IEV2, I, XC, X, KI) calculates the centroid of complex.

APPENDIX - A

FEATURES OF COMPUTER PROGRAMS

SUBROUTINE FUNC (N,M,K, X,F,I) Specifies the objective function (user supplied)

SUBROUTINE BOUND (N,M,K,X,G,H,I) specifies the explicit constraints with upper and lower bounds

SUBROUTINE CONST (N,M,K,X,G,H,I) specifies the implicit constraints with upper and lower bounds

SUBROUTINE RANDOM (N,K,FF,R) generates the random numbers.

A 1.1.3 DESCRIPTION OF PARAMETERS

N Number of independent variables defined in main program

M Number of constraints defined in main program

K Number of points in the complex defined in the main program

NMAX Maximum number of iterations defined in main program

ALPHA Reflection factor defined in main program

EPS1 A small quantity to compare with ALPHA-defined in main program

BETA Convergence parameter defined in main program

GAMMA Convergence parameter defined in main program

DELTA Explicit constraint violation correction defined in main program

IPRINT IPRINT=1, PRINT FINAL RESULT

IPRINT=0, PRINT INTERMEDIATE RESULT, defined in main program

X Independent variable defined initial values in main program

R Randam numbers between 0 and 1 defined in subroutine RANDOM

F Objective function defined in subroutine FUNC

IT Iteration index - defined in subroutine CONSX

IEV2 Index of point with maximum function value defined in subroutine CONSX

IEV1 Index of point with minimum function value defined in subroutine CONSX

G Lower constraint defined in subroutine BOUND and CONST respectively

H Upper constraint defined in subroutine BOUND and CONST

XC Centroid defined in subroutine CENTR

I Point Index -defined in subroutine CONSX

KODE Key used to determine if implicit constraints are provided - defined in subroutine CONSX and CHECK

K1 Do loop limit defined in subroutine CONSX

FF 4 digit number to generate random number defined in main program.

A 1.1.4 INPUT

N,M,K, NMAX, IPRINT, ALPHA, BETA, GAMMA, and (X(J,1), J=1,N)

A 1.1.5 OUTPUT

The main program first prints out values of all input parameters.

Subroutine CONSX provides intermediate output on each iteration provided the user specfies iprint=1, If Iprint=0, only final result is printed.

When the solution has converged to within the allowable range, or when maximum number of iterations has been exceeded.

the main program prints the final vlaue of the function, the X vector and the total number of function evaluations and centroid of complex.

A 1.1.6 SUMMARY OF USER REQUIREMENTS

(a) Determine the value for N,M,K, NMAX, ALPHA, BETA, GAMMA, DELTA, IPRINT. Guide lines for specifying the parameters are as follows

K = N + 1

ALPHA = 1.3

BETA = some small number, say magnitude of function time 10^{-4}

GAMMA = 5

DELTA = Some small number, say magnitude of order X vector times 10^{-4}

(b) Determine the initial estimates for optimum values of independent variables :

enter as (X(J,1), J=1,N)

(c) Supply the value of FF (4 digit number) to generate the random numbers

(d) Adjust DIMENSION and FORMAT statement as necessary

(e) Specify objective function by writing SUBROUTINE FUNC

(f) Define H (upper bounds) and G (lower bounds) explicit constraints in subroutine BOUND and implicit constraints in subroutine CONST.

A 2. MARQUARDT METHOD

A 2.1 PROGRAM DESCRIPTION

A 2.1.1 USAGE

The program consists of a main program a general subroutine BSOLVE, a general function subprogram ARCOS, and two user supplied subroutines FUNC and DERIV. All input and output is through the main program. Format changes may be required depending on the problem under consideration.

A 2.1.2 SUBROUTINES REQUIRED

SUBROUTINE BSOLVE (KK,B, NN, Z, Y, PM, ITER, FLA, TAU, EPS, PHMIN, I, ICON, FV, DV, BV, BMIN, BMAX, P, FUNC, DERIV, KD, A, AC, GAMM) called from main program - performs primary calculation and coordinates other subroutines.

SUBROUTINE (KK,B, NN, Z, PJ, FV, DV, J, JTEST) specifies analytical derivaties if used, omit if numerical derivative used. (user supplied).

SUBROUTINE FUNC (KK, B, NN, Z, FV) specifies the model (user supplies).

FUNCTION ARCOS(Z) general function subprogram internal to BSOLVE.

A 2.1.3 DESCRIPTION OF PARAMETERS

NN = Number of data points or number of equations

KK = Number of unknowns

B = Vector of unknowns

BMIN = Vector of minimum values of B

BMAX = Vector of maximum values of B

X = Vector of independent variable data point

Y = Vector of dependent variables

PH = Least square objective function

Z = Computed values of dependent variables

BV = Code vector set equal to 1 for numerical derivatives and -1 for analytical derivatives

ITER = Iterations.

A 2.1.4 DIMENSIONS REQUIREMENTS

The dimension STATEMENT IN THE MAIN program and subroutines should be modified according to requirements of each particular problem.

A 2.1.5 INPUTS

(B(J),J=1, KK), (BMIN(J), J=1,KK)

(BMAX(J), J=1,KK), (X(I),I=1,NN), (Y(I),I=1,NN)

A 2.1.6 OUTPUT

The main program prints out all input data the values of unknowns, least square function value (PH), and function evaluations at each iteration, in addition to final values of unknowns.

A 2.1.7 SUMMARY OF USER REQUIRMENTS

1. Determine the value for NN, KK, B(J), BMIN(J), BMAX(J), X(I), Y(I).
2. Adjust the dimension statements in main program and subroutines.

3. Specify analytical derivative in DERIV, if used

4. Change the input and output format statement as necessary.

5. Specify the model in FUNC subroutine.

A 3. INTERIOR PENALTY FUNCTION METHOD

A 3.1 PROGRAM DESCRIPTION

A 3.1.1 USAGE

Program consists of a main program and three gereal subroutines UNCON, ONEDIM, GRADT and two user's supplied subroutine FTN and CONST. All input and output is through the main program, format may by changed depending upon the problem under consideration.

A 3.1.2 SUBROUTINES REQUIRED

SUBROUTINE UNCON(N,M,R,EPS, EPSS, STEPO, X, XOPT, GRAD, F, OBJ, IT, KT) solves the unconstrained minimization problem.

SUBROUTINE ONEDIM(N, M,R, EPSS, STEPO, SLAMDA, X, XN, SS, F, FN, OBJ, GRAD, GRADN) specifies the optimum step length in current search direction.

SUBROUTINE GRADT (X, N, M, R, GRAD, FF) calculates the derivatives numerically using Newton's forward formula.

SUBROUTINE FTN (X, F, OBJ, N, M, R) specifies the function (user's supplied)

SUBROUTINE CONST (N, X, VAL, IN) specifies the constraints (user's supplied)

A 3.1.3 DESCRIPTION OF PARAMETERS

N = number of variables

M = number of constraints

MAXPI = maximum number of ϕ function to be minimized.

ITLIM = maximum number of iterations permitted in any one unconstrained minimization.

MAXIN = maximum number of cubic interpolation permitted in any one dimensional search.

SLAMDA = optimum step length.

C1 = reduction factor for the penalty parameter.

R = initial value of penalty parameter.

EPS = convergence required in unconstrained minimization.

EPSS = convergence required in cubic interpolation.

STEPO = trial step length to be used in cubic interpolation.

X(I) = independent design variables.

OBJ = value of objective function.

F = value of ϕ function.

SS(I) = component of search direction.

IT = iteration number in unconstrained minimization.

KT = iteration number in main program.

GRAD = Component of gradient vector.

A 3.1.4 DIMENSION REQUIREMENT

The DIMENSION statements in the main the program and subroutines should be modified according to problem under consideration.

A 3.1.5 INPUT

N, M, R, C1, (X(I), I=1,N), STEPO, MAXPI, ITLIM, MAXIN, EPS, EPSS.

A 3.1.6 OUTPUT

The main program prints out all input data and values of main program iteration number starting values of ϕ function and variables before starting the unconstrained minimization optimum values of variables after unconstrained iteration. The value of objective function and ϕ function, functions evaluations at each main program iteration and final values of total function evaluation, final values of variables and objective function.

A 3.1.7 SUMMARY OF USER'S REQUIREMENT

1. Feed the values of all input data.
2. Adjust dimension statements in main program and subroutines.
3. specify the function in FTN subroutine.
4. specify the constraints in CONST subroutine.
5. change the input and output format statement as necessary..

APPENDIX – B

EXPERIMENTAL DATA POINTS

APPENDIX B

SIMULATED EXPERIMENTAL DATA POINTS

TABLE - B1

Temp. of reaction (K)	Mole fraction of ethylene	Mole fraction of hydrogen
45.0	.35	.65
53.0	.30	.70
60.0	.28	.72
68.0	.38	.45
75.0	.25	.55
85.0	.15	.60
90.0	.10	.40
85.80	.10	.80
67.43	.10	.80
90.0	.10	.40
67.82	.10	.80
67.81	.10	.80
90.00	.10	.40
82.09	.40	.60
80.82	.20	.80
90.0	.40	.42
80.82	.20	.80
80.82	.20	.80
90.0	.40	.42
85.80	.10	.80
82.09	.40	.60
65.58	.40	.60
65.62	.40	.60
80.82	.20	.80