GENERAL PROGRAMS FOR

LEAST pTH AND NEAR MINIMAX APPROXIMATION

GENERAL PROGRAMS FOR
LEAST pTH AND NEAR MINIMAX APPROXIMATION

by

Jadranka K. Rizoniko - Popović, Dipl. Eng.

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Engineering

McMaster University

October 1972

MASTER OF ENGINEERING (1972)
(Electrical Engineering)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: General Programs for Least pth and Near Minimax Approximation

AUTHOR: Jadranka K. Rizoniko-Popović, Dipl. Eng. (University of Belgrade)

SUPERVISOR: J. W. Bandler, B.Sc. (Eng.), Ph.D. (University of London), D.I.C. (Imperial College)

NUMBER OF PAGES: vii, 144.

SCOPE AND CONTENTS:

User-oriented computer programs in FORTRAN IV for discrete least pth approximation with a single specified function, and more generalized discrete least pth approximation with various specifications, which may also be used for nonlinear programming, are presented. Values of p up to 10^6 can be used successfully in conjunction with efficient gradient minimization algorithms such as the Fletcher-Powell method and a recent method due to Fletcher. It has already been demonstrated how efficiently extremely near minimax results can be achieved on a discrete set of sample points using this approach and the programs written verify this. The programs may be applied to a wide variety of design problems with a wide range of specifications. They are suitable for electrical network and system design and such problems as filter design.

ACKNOWLEDGEMENTS

The author wishes to express appreciation to Dr. J. W. Bandler for his guidance and encouragement in the preparation of this Thesis. The author is also sincerely thankful to C. Charalambous and other colleagues for fruitful discussions and help.

Financial support provided by the National Research Council of Canada through grants A7239 and C154 and by the Department of Electrical Engineering at McMaster University in the form of a teaching assistantship is gratefully acknowledged.

I would like to thank Mary Heatherington for typing the manuscript.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

Introduction

Two complete user-oriented computer programs in FORTRAN IV are presented which utilize some new ideas on discrete least pth approximation [1] and, more recent ideas on generalized least pth approximation [2]. Least pth approximation with p=2 gives a discrete least squares approximation. With sufficiently large values of p an optimal solution very close to the optimal minimax solution can be obtained. Values of p up to 10^6 have been successfully employed. Gradient minimization algorithms due to Fletcher and Powell and, more recently, to Fletcher are used. The user has to write all the required specifications, the approximating functions and weighting functions in a straightforward way.

The first program is described in Chapter V with a listing attached in the Appendices and is applicable to design problems with a single specification. Quadratic interpolation, if desired, is employed to bring the discrete approximation solution closer to the solution of the continuous approximation problem. Numerical examples for which the minimax solutions are known were chosen to illustrate the work of the program. The solutions obtained are in excellent agreement with the known ones.

The second program is described in Chapter VI with instructions as to its use and the listing being given in the Appendices. The

program is directly applicable to such problems as meeting or exceeding design specifications on several disjoint closed intervals as in filter design and allows for situations more general than the conventional problem of approximating a single continuous function on a closed interval. There is no restriction on the number of variable parameters, discrete point sets and number of intervals. The examples which demonstrate that the program works, since the methods have already been tested, were chosen in system modelling and multi-section lossless transmission-line network design. Although the program is not written for nonlinear programming we found it is also applicable for problems with parameter constraints.

The programs are run on the CDC 6400 at the McMaster University Computer Center.

CHAPTER II

The Optimization Problem

The general optimization problem is to maximize or minimize a function $f(\underset{\sim}{a})$, where

$$\underset{\sim}{a} \triangleq \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_k \end{bmatrix} \tag{2.1}$$

is a vector of k independent parameters or design variables. Since the problem of maximizing $f(\underset{\sim}{a})$ is equivalent to that of minimizing $-f(\underset{\sim}{a})$ only the minimization problem is considered. The function to be minimized is called the objective function and will be denoted by the symbol U.

In general, a linear or nonlinear set of constraints may have to be satisfied either during the optimization process or by the optimal solution. Each parameter might be constrained by an upper and lower bound

$$\underset{\sim}{a}_\ell \leq \underset{\sim}{a} \leq \underset{\sim}{a}_u \tag{2.2}$$

where

(2.3) $$\underset{\sim}{a}_{\ell} \triangleq \begin{bmatrix} a_{\ell 1} \\ a_{\ell 2} \\ . \\ . \\ . \\ a_{\ell k} \end{bmatrix} \quad \text{and} \quad \underset{\sim}{a}_{u} \triangleq \begin{bmatrix} a_{u1} \\ a_{u2} \\ . \\ . \\ . \\ a_{uk} \end{bmatrix}$$

are lower and upper bounds, respectively. A possible set of h implicit constraint functions may be set as

(2.4) $$\underset{\sim}{c}(\underset{\sim}{a}) \geq \underset{\sim}{0}$$

where

(2.5) $$\underset{\sim}{c} \triangleq \begin{bmatrix} c_1 (\underset{\sim}{a}) \\ c_2 (\underset{\sim}{a}) \\ . \\ . \\ . \\ c_h (\underset{\sim}{a}) \end{bmatrix}$$

and $\underset{\sim}{0}$ is a zero vector. Any vector $\underset{\sim}{a}$ which satisfies the constraints is called feasible. A region of feasible points $\underset{\sim}{a}$ for which (2.2) and (2.4) are satisfied is a feasible region R.

The problem of minimizing U subject to the constraints is to locate a point $\overset{v}{\underset{\sim}{a}}$ such that

(2.6) $$\overset{v}{U} \triangleq f(\overset{v}{\underset{\sim}{a}}) = \min_{\underset{\sim}{a} \in R} f(\underset{\sim}{a}) \leq f(\underset{\sim}{a})$$

for any feasible $\underset{\sim}{a}$ in the neighborhood of $\overset{v}{\underset{\sim}{a}}$. A point $\overset{v}{\underset{\sim}{a}}$ is called a constrained local minimum.

Methods which guarantee convergence to a global minimum for all functions are not available. Since we cannot generally guarantee to find a global minimum we restrict our discussion to local mimima. The distinction between local and global minima is not essential from the optimization point of view; it is,however, important when the results of applying optimization techniques have to be interpreted.

Most optimization strategies reduce a constrained problem into an essentially unconstrained one. This can be accomplished by transforming the parameters and leaving the objective function unaltered [3], [4], or by modifying the objective function by introducing some kind of penalty [5] - [8]. Other methods for handling constraints employ changes in strategy when constraint violations occur, but they are not found to be so efficient and accurate. Further details may be found in some of the general references [8] - [11].

CHAPTER III

Methods of Minimization

3.1 Direct search methods

Many direct search methods for optimization were proposed between 1960 and 1965. Direct search methods are based on a sequential examination of trial solutions which, by simple comparisons, give an indication for a further searching procedure. These methods require only the evaluation of the function at a given point and thus are applicable to general continuous functions. In general, they do not give a rapid rate of ultimate convergence and hence are inefficient for finding a minimum. Methods of this type are useful in the early stages of optimization and can provide information about a region in which a minimum is located. Many direct search methods for unconstrained optimization are described by Kowalik and Osborne [12] and by Box et al [13].

In the last few years, however, relatively few papers have been published on algorithms that do not require the calculation of any derivatives. The main subjects of these recent papers are estimating derivatives by numerical differences in order to use Davidon's [14] algorithm, lattice approximations, and fitting a quadratic function to calculate function values.

3.2 Gradient methods of minimization

In this section it is assumed that appropriate partial derivatives exist and methods are described which utilize partial derivative information to determine the direction of search for a minimum. The partial derivatives may be defined analytically at each point or must be estimated from the differences of values of $f(\underset{\sim}{a})$.

For a continuous objective function $f(\underset{\sim}{a})$ with continuous first and second derivatives, we can write, using Taylor's theorem

$$f(\underset{\sim}{a}+\Delta\underset{\sim}{a}) = f(\underset{\sim}{a})+(\underset{\sim}{\nabla}f)^T\Delta\underset{\sim}{a}+\tfrac{1}{2}\Delta\underset{\sim}{a}^T\underset{\sim}{G}(\underset{\sim}{a}+\lambda\Delta\underset{\sim}{a})\Delta\underset{\sim}{a} \tag{3.1}$$

where $0 \leq \lambda \leq 1$,

$$\Delta\underset{\sim}{a} \triangleq \begin{bmatrix} \Delta a_1 \\ \Delta a_2 \\ . \\ . \\ . \\ \Delta a_k \end{bmatrix} \tag{3.2}$$

represents an incremental vector,

$$\underset{\sim}{\nabla}f \triangleq \begin{bmatrix} \frac{\partial f}{\partial a_1} \\ \frac{\partial f}{\partial a_2} \\ . \\ . \\ . \\ \frac{\partial f}{\partial a_k} \end{bmatrix} \tag{3.3}$$

is the gradient vector containing the first partial derivatives and

$$(3.4) \qquad \underset{\sim}{G} \triangleq \underset{\sim}{G}(\underset{\sim}{a}) \triangleq \begin{bmatrix} \frac{\partial^2 f}{\partial a_1^2} & \frac{\partial^2 f}{\partial a_1 \partial a_2} & \cdots & \frac{\partial^2 f}{\partial a_1 \partial a_k} \\ \frac{\partial^2 f}{\partial a_2 \partial a_1} & \cdots & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \\ \frac{\partial^2 f}{\partial a_k \partial a_1} & \cdots & & \frac{\partial^2 f}{\partial a_k^2} \end{bmatrix}$$

is the matrix of second partial derivatives, the Hessian matrix. For twice continuously differentiable functions $\underset{\sim}{G}$ exists and is symmetric, i.e., $\underset{\sim}{G}^T = \underset{\sim}{G}$. A symmetric matrix $\underset{\sim}{G}$ is positive definite if for all $\underset{\sim}{a} \neq \underset{\sim}{0}$

$$(3.5) \qquad \underset{\sim}{a}^T \underset{\sim}{G}\, \underset{\sim}{a} > 0.$$

If $\underset{\sim}{G}$ is positive definite for all $\underset{\sim}{x}$ then according to (3.1)

$$(3.6) \qquad f(\underset{\sim}{a}+\Delta\underset{\sim}{a}) > f(\underset{\sim}{a})+(\underset{\sim}{\nabla} f)^T \Delta\underset{\sim}{a}$$

for all $\underset{\sim}{a}$. At a local minimum

$$(3.7) \qquad \underset{\sim}{\nabla} f(\overset{v}{\underset{\sim}{a}}) = \underset{\sim}{0}$$

must hold. Assuming the first and second derivatives exist, a point is a minimum if the gradient vector is zero and the Hession matrix is positive definite at that point.

Since the first partial derivatives of a function vanish at the minimum the second-order terms in the Taylor series expansion dominate in the vicinity of the minimum. It follows that the only methods which will minimize a general function quickly and efficiently are those which work well on a quadratic form and are guaranteed to converge eventually for a general function. All others may be slow, at least in the vicinity of the minimum, and often elsewhere.

The general quadratic function of k variables can be written

$$Q(\underset{\sim}{a}) = c + \underset{\sim}{b}^T\underset{\sim}{a} + \tfrac{1}{2}\underset{\sim}{a}^T\underset{\sim}{A}\underset{\sim}{a} \tag{3.8}$$

with gradients

$$\underset{\sim}{g} = \underset{\sim}{b} + \underset{\sim}{A}\underset{\sim}{a} \tag{3.9}$$

where $\underset{\sim}{A}$ is a kxk symmetric matrix, and c and $\underset{\sim}{b}$ are constants. If $\underset{\sim}{A}$ is positive definite

$$\overset{v}{\underset{\sim}{a}} = -\underset{\sim}{A}^{-1}\underset{\sim}{b} \tag{3.10}$$

and $\overset{v}{\underset{\sim}{a}}$ is unique and can be found in a finite number of steps.

We are interested in a quadratic approximation, such as (3.8), to the objective function $f(\underset{\sim}{a})$, because a quadratic is the simplest differentiable function that has a well-defined minimum. In this case in the ith iteration the obvious quadratic approximation to $f(\underset{\sim}{a})$ is

$$f(\underset{\sim}{a}) \approx f(\underset{\sim}{a}_i) + \underset{\sim}{g}_i^T\Delta\underset{\sim}{a}_i + \tfrac{1}{2}(\Delta\underset{\sim}{a}_i)^T\underset{\sim}{G}_i\Delta\underset{\sim}{a}_i \tag{3.11}$$

where $\underset{\sim}{g}_i$ is the first derivative vector and $\underset{\sim}{G}_i$ is the second

derivative matrix of $f(\underset{\sim}{a})$ at point $\underset{\sim}{a}_i$. The iterative process known as Newton iterative procedure is defined by the equation

$$\underset{\sim}{a}_{i+1} = \underset{\sim}{a}_i - (\underset{\sim}{G}_i)^{-1} \underset{\sim}{g}_i . \tag{3.12}$$

The generalized Newton-Raphson method [15] has fast convergence eventually, but frequently fails to converge from a poor initial estimate of the minimum.

A quasi-Newton iterative algorithm that involves linear searches does not require second-order derivatives of the function to be evaluated [16]. The term quasi-Newton means that an attempt is made to simulate equation (3.12), and a matrix $\underset{\sim}{H}_i$ is used, which is an estimate of $(\underset{\sim}{G}_i)^{-1}$. Because linear searches are made, $\underset{\sim}{a}_{i+1}$ may be defined by the formula

$$\underset{\sim}{a}_{i+1} = \underset{\sim}{a}_i - \lambda_i \underset{\sim}{H}_i \underset{\sim}{g}_i \quad , \tag{3.13}$$

where λ_i is a multiple which is calculated at each iteration.

The recent papers on minimization algorithms propose modifications to the iteration (3.12), so that it is not necessary for $\underset{\sim}{a}_0$ to be close to the solution. Gradient methods that are used at the present time are summarized in a general theory published in reference [17].

The ideas resulting from equation (3.13) have been successfully employed in an iterative procedure in the gradient optimization methods of Fletcher and Powell [18]. Generally acknowledged to be among the most powerful minimization methods currently available,

when first derivatives are analytically defined, is this method and a more recent one due to Fletcher [19]. As they are both used in a program presented in this thesis, a brief discussion of the methods follows.

3.3 The Fletcher-Powell method

The method presented by Fletcher and Powell is an iterative method for finding a local minimum of a general function $f(\underset{\sim}{a})$ where the gradient vector $\underset{\sim}{g}$ (3.9) is defined analytically. It is based upon Davidon's variable metric method [4].

In this method the matrix $\underset{\sim}{G}$ from (3.8) is not evaluated directly but instead another matrix $\underset{\sim}{H}$ is introduced and used as an approximation to $\underset{\sim}{G}^{-1}$. The increment $\underset{\sim}{\delta}$ is taken along the direction of search $\underset{\sim}{s}$

$$\underset{\sim}{s} = -\underset{\sim}{H}\underset{\sim}{g} \tag{3.14}$$

so that

$$\underset{\sim}{\delta} = \alpha\underset{\sim}{s} \tag{3.15}$$

where α is such that $f(\underset{\sim}{a}+\lambda\underset{\sim}{s})$ is a minimum with respect to λ along $\underset{\sim}{a} + \lambda\underset{\sim}{s}$. The approximating matrix $\underset{\sim}{H}$ is updated at each iteration using the formula

$$\underset{\sim}{H}_{i+1} = \underset{\sim}{H}_i + \frac{\underset{\sim}{\delta}_i\underset{\sim}{\delta}_i^T}{\underset{\sim}{\delta}_i^T\underset{\sim}{\gamma}_i} - \frac{\underset{\sim}{H}_i\underset{\sim}{\gamma}_i\underset{\sim}{\gamma}_i^T\underset{\sim}{H}_i}{\underset{\sim}{\gamma}_i^T\underset{\sim}{H}_i\underset{\sim}{\gamma}_i} \tag{3.16}$$

where

$$\underset{\sim}{\delta}_i = \underset{\sim}{a}_{i+1} - \underset{\sim}{a}_i = -\alpha_i \underset{\sim}{H}_i \underset{\sim}{g}_i \tag{3.17}$$

and

$$\underset{\sim}{\gamma}_i = \underset{\sim}{g}_{i+1} - \underset{\sim}{g}_i \tag{3.18}$$

are the changes in $\underset{\sim}{a}$ and $\underset{\sim}{g}$ during the ith iteration. If $\underset{\sim}{H}_0$ is positive definite initially then it can be proved inductively that all subsequent $\underset{\sim}{H}_i$ are also positive definite. Since $\underset{\sim}{H}_0$ is initially chosen as the identity matrix, then all $\underset{\sim}{H}_i$ will be positive definite. As a consequence the method converges. When the method is applied to the quadratic function it leads to a minimum in k iterations and it is further proved that the matrix $\underset{\sim}{H}_k$ converges to the inverse of the Hessian matrix $\underset{\sim}{G}^{-1}$.

The algorithm needs to find the multiple α at each iteration. A sufficiently accurate minimum is obtained by evaluating the function and gradient for different values of λ and using cubic interpolation. The linear search usually requires considerable computing effort.

It is practicable to apply this method to find a local minimum of a general function whose first derivatives can be evaluated quickly,even if only a poor initial approximation to a solution is known. Both the proof of convergence and success in practice depend on accurate location of the minimum in the linear searches.

The termination criteria for the Fletcher-Powell iterative procedure are as follows:

1. If the function value has not decreased in the last iteration step, the search for the minimum is terminated provided the gradient vector is already sufficiently small.

2. If the elements of vector $\underset{\sim}{a}$ change by very small amounts, and at least k iterations are performed, the minimization is terminated.

3. If the number of iterations exceeds an upper bound preassigned by the user, further computation is bypassed, and the appropriate message indicates poor convergence.

4. If one of the successive linear searches indicates that no minimum exists, further calculation is bypassed and the message indicates that no minimum exists.

3.4 The Fletcher method

A method presented by Fletcher came as a result of further development of the Fletcher-Powell method. He retains the positive definiteness of $\underset{\sim}{H}$ in the new algorithm, as it ensures a reduction in the function value on each iteration, but a multiple α is not calculated at a point where the function is minimized locally along the direction of search.

The idea of replacing the linear search, which is time consuming, by a simpler way of reducing the function at each iteration

has the consequence of abandoning the property of quadratic termination which guarantees fast ultimate convergence. If linear search is abandoned, something else has to force a sufficiently large decrease in $f(\underset{\sim}{a})$ at each iteration to guarantee convergence. The change in function in one iteration would be expected by Taylor's series to be approximately $\underset{\sim}{g}^T\underset{\sim}{\delta}$ when $\underset{\sim}{\delta}$ is small, but much less than that in absolute value when the position of the minimum along the line is overestimated. When that happens, the change in $f(\underset{\sim}{a})$ relative to $\underset{\sim}{g}^T\underset{\sim}{\delta}$, i.e. $\Delta f/\underset{\sim}{g}^T\underset{\sim}{\delta}$, cannot become arbitrarily small.

The corrections are determined by

$$\underset{\sim}{\delta} = -\lambda \underset{\sim}{H}\underset{\sim}{g} \tag{3.19}$$

and trying the values of $\lambda = 1, w, w^2, \ldots$ for $0<w<1$ will eventually produce a $\underset{\sim}{\delta}$ which satisfies the preassigned ratio $\Delta f/\underset{\sim}{g}^T\underset{\sim}{\delta}$.

Another important feature is to retain a strict positive definiteness of the approximating matrices if these ideas are to be used. Failure may be caused by slow rate of convergence and due to round-off errors in $\underset{\sim}{H}$ which may become singular. So a new formula for updating $\underset{\sim}{H}$ was found

$$\underset{\sim}{H}_{i+1} = \underset{\sim}{H}_i - \frac{\underset{\sim}{\delta}_i\underset{\sim}{\gamma}_i^T\underset{\sim}{H}_i}{\underset{\sim}{\delta}_i^T\underset{\sim}{\gamma}_i} - \frac{\underset{\sim}{H}_i\underset{\sim}{\gamma}_i\underset{\sim}{\delta}_i^T}{\underset{\sim}{\delta}_i^T\underset{\sim}{\gamma}_i} + \left(1 + \frac{\underset{\sim}{\gamma}_i^T\underset{\sim}{H}_i\underset{\sim}{\gamma}_i}{\underset{\sim}{\delta}_i^T\underset{\sim}{\gamma}_i}\right)\frac{\underset{\sim}{\delta}_i\underset{\sim}{\delta}_i^T}{\underset{\sim}{\delta}_i^T\underset{\sim}{\gamma}_i} \tag{3.20}$$

where $\underset{\sim}{\delta}_i$ and $\underset{\sim}{\gamma}_i$ are defined in (3.17) and (3.18), respectively.

The algorithm takes into account the above discussion on $\underset{\sim}{H}$ and λ. It is desirable to have some guarantee that the $\underset{\sim}{H}$ matrices tend to $\underset{\sim}{G}^{-1}$ so that the ultimate convergence for quadratic functions can be proved. A suitable property refered as a Property 1 has been defined in [19] which requires that, for quadratic functions, the eigen values of $\underset{\sim}{H}$ must tend monotonically to those of $\underset{\sim}{G}^{-1}$ in a certain sense. The formula (3.20) possesses the required Property 1. It is shown that the formula (3.16) also possesses that property [19]. The two formulae can be combined together and they can be used to minimize quadratic functions and thus permit as close an approximation to $\underset{\sim}{G}^{-1}$ as possible. The test which predicts which formula will be used is the following

$$\underset{\sim}{\delta}^T\underset{\sim}{\gamma} \geq \underset{\sim}{\gamma}^T\underset{\sim}{H}\underset{\sim}{\gamma} \quad . \tag{3.21}$$

If (3.21) is true the formula (3.20) is used, if not the formula (3.16) is used. In fact when equality holds no preference is important but it was decided to use the formula (3.20) to avoid possible singularity in $\underset{\sim}{H}$.

In most cases for $\lambda = 1$ convergence will occur and so the method requires only one evaluation of f and $\underset{\sim}{g}$ per iteration which results in the efficiency of the algorithm. If $\lambda = 1$ is not successful in reducing f sufficiently, then a λ is determined by cubic interpolation and the largest of this value and 0.1 is used. Thus the possibility of obtaining a local minimum along a line is retained when it is necessary, in which case more than one evaluation of f and

$\underset{\sim}{g}$ is used in one iteration. The property of quadratic termination has been replaced by a property in which the approximating matrix $\underset{\sim}{H}$ has to tend to the inverse Hessian $\underset{\sim}{G}^{-1}$ in a certain sense.

The termination criteria for the Fletcher iterative procedure are based on the same ideas used for the termination of the Fletcher-Powell method.

CHAPTER IV

Some Design Objectives

4.1 Error criterion

Most electrical network design problems can be formulated as approximation problems. Evaluation of the difference between a specified function and an approximating function leads to an error criterion. Let us define a weighted error function or deviation between a specified function and an approximating function as

$$e(\underset{\sim}{a},x) \triangleq w(x)(F(\underset{\sim}{a},x) - S(x)) \tag{4.1}$$

where

$S(x)$	is the real or complex specified function,
$F(\underset{\sim}{a},x)$	is the real or complex approximating function,
$w(x)$	is the positive weighting function,
$\underset{\sim}{a}$	is a k dimensional vector representing the adjustable parameters.

The approximation problem when all the functions from (4.1) are continuous and real is illustrated on Fig. 4-1.

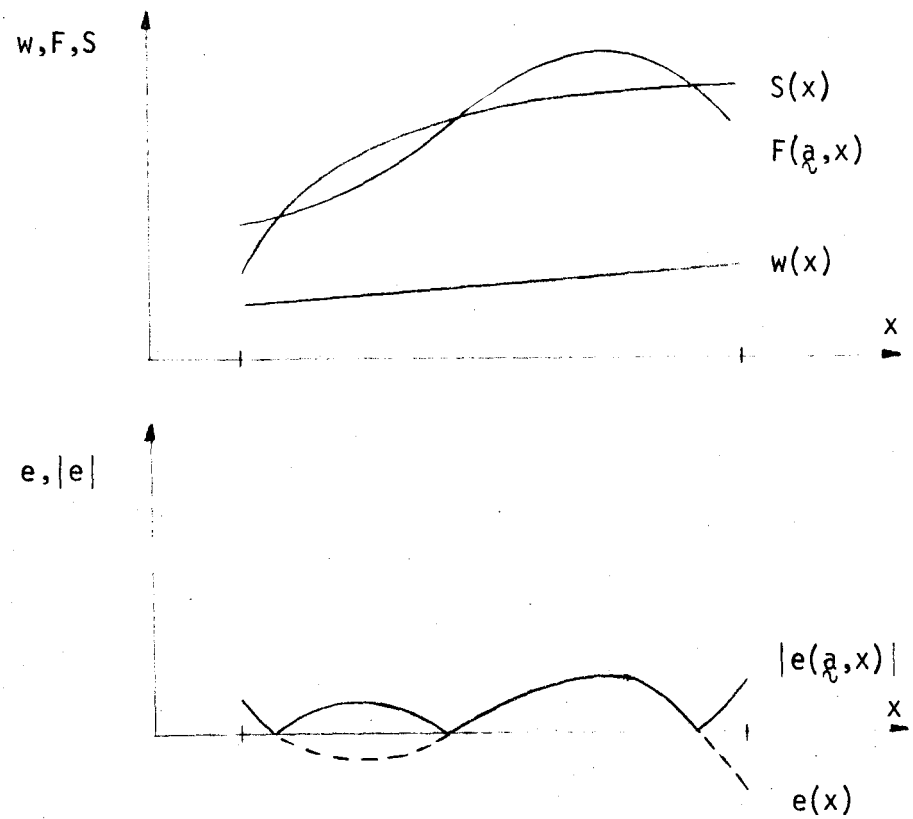


Fig. 4-1 Continuous approximation problem

In general, more can be done for a finite point set of independent variables than for the whole interval. With a discrete set some values of the independent variables can be isolated or approximation can be done on several disjoint closed intervals as is shown in Fig. 4-2. Of course, in a case when a curve has to be fitted to experimental data discrete approximation may be the only

approach possible. It is appropriate then to consider discrete approximation.

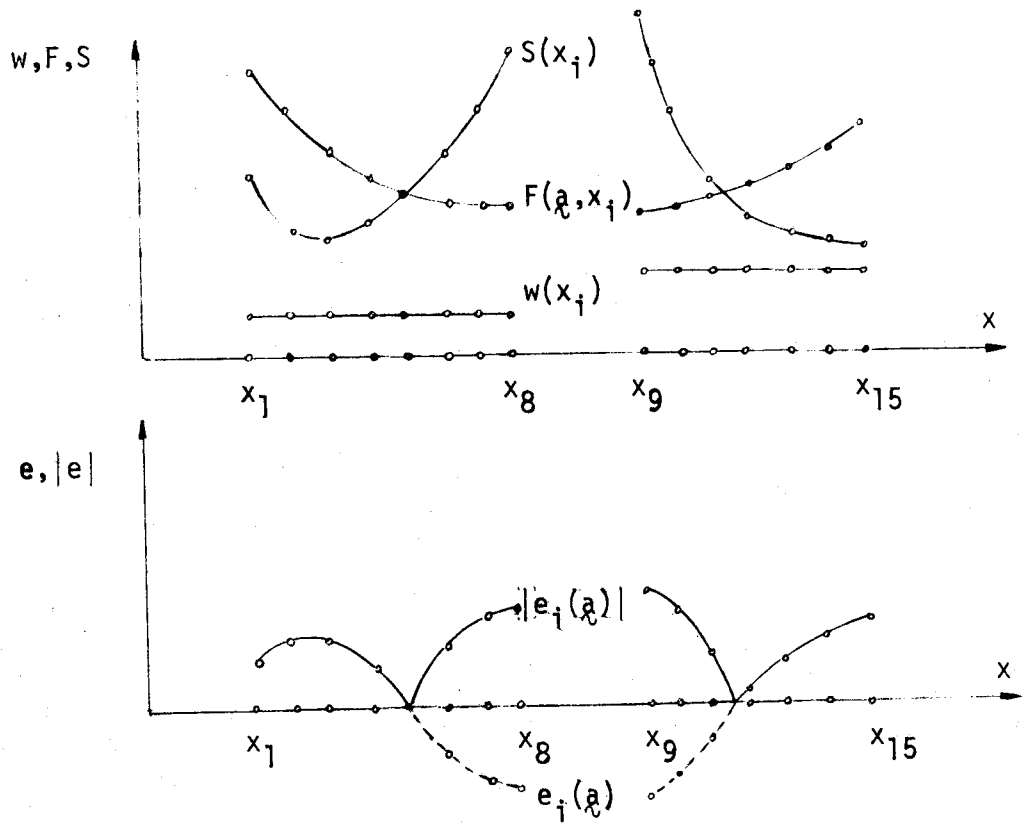


Fig. 4-2 Discrete approximation problem

In many problems of interest a relatively modest number of well-chosen sample points result in solutions to the design problem sufficiently good for all practical purposes.

A discrete version of the error function (4.1) is given by

$$e_i(\underset{\sim}{a}) \triangleq e(\underset{\sim}{a},x_i) = w(x_i)(F(\underset{\sim}{a},x_i)-S(x_i)) \quad , \; i \in I \tag{4.2}$$

where I is a sample set of n points.

A purpose of the weighting function w(x) is to emphasize or deemphasize various parts of the difference between the approximating function and the specification to suit the designer's requirement. Thus the optimum with respect to one weighting function may not be an optimum with respect to another.

4.2 L_p and $\ell_p(n)$ norms

We may define a norm on a closed interval $[x_a, x_b]$

$$||e||_p \triangleq \left\{ \int_{x_a}^{x_b} |e(\underset{\sim}{a},x)|^p \, dx \right\}^{\frac{1}{p}} \qquad p \geq 1 \tag{4.3}$$

called the L_p norm for the continuous case, and a norm

$$||\underset{\sim}{e}||_p^n \triangleq \left\{ \sum_{i \in I} |e_i(\underset{\sim}{a})|^p \right\}^{\frac{1}{p}} \qquad p \geq 1 \tag{4.4}$$

where

$$\underset{\sim}{e}(\underset{\sim}{a}) \triangleq \begin{bmatrix} e_1(\underset{\sim}{a}) \\ e_2(\underset{\sim}{a}) \\ . \\ . \\ . \\ e_n(\underset{\sim}{a}) \end{bmatrix} \tag{4.5}$$

called the $\ell_p(n)$ norm as a discrete approximation to (4.3). The process of minimizing the objective function defined as the L_p

or $\ell_p(n)$ norm is called least pth approximation. The solution is termed a best approximation with respect to $||e||_p$ for the continuous or $||\underset{\sim}{e}||_p^n$ for the discrete case. A widely known case is for p = 2, namely least squares approximation. An important limiting case where p approaches infinity is minimax, Chebyshev or L_∞ and $\ell_\infty(n)$ approximation. In this case we may define

$$(4.6) \qquad ||e||_\infty \triangleq \max_{[x_a, x_b]} |e(\underset{\sim}{a}, x)|$$

and

$$(4.7) \qquad ||\underset{\sim}{e}||_\infty^n \triangleq \max_{i \in I} |e_i(\underset{\sim}{a})|$$

the Chebyshev or uniform norms.

In network and system design, the optimal responses usually turn out to be equal-ripple, although equal-ripple responses do not necessarily yield optimal solutions in the minimax sense.

It remains to point out that the optimization of the objective functions when defined as the uniform norms, by any of the well known methods for differentiable functions is unsatisfactory because such methods attempt to reduce the norm of the gradient function to zero whereas in the minimax problem, the gradient may be discontinuous.

In spite of this some successful algorithms and computer programs with objectives in the form of (4.6) and (4.7) for a certain class of functions have been reported [20] - [22] and a good survey of

practical minimax approximation is presented in [23].

Least pth approximation does not suffer from this difficulty of minimax objectives. Therefore least pth approximation with values of p, $1<p<\infty$, has been proposed and used to obtain optimal designs which produce the results close to the minimax reponse. For the special error criterion of (4.4) where p = 2, a simplified method has been developed [3], [24] and [25]. Temes and Zai [26] and [27] have extended a least squares method to a least pth method using for minimization in place of $||\underset{\sim}{e}||_p^n$ the following objective function

$$U = \sum_{i \in I} (e_i(\underset{\sim}{a}))^p \text{ for } p \geq 2 \text{ and } p \text{ even.} \tag{4.8}$$

They developed an algorithm for least pth approximation [27] usable with moderately large values of p, say of the order of 10. Afterwards, values of p up to 10^{12} have been successfully employed minimizing the $\ell_p(n)$ norm in conjunction with efficient gradient minimization algorithms such as the Fletcher-Powell method and the recent method due to Fletcher [1]. Bandler and Charalambous generalized the objectives so that least pth approximation could be used to meet or exceed circuit or system performancy specifications [2].

More recent theoretical work has been published on conditions for optimality in least pth approximation with $p \to \infty$ [28] from which conditions for a minimax approximation [29] fall out.

CHAPTER V

Practical Least pth Approximation

5.1 Scaling

The obvious reason why large values of p are desirable in the least pth objective is that the corresponding optimal approximations tend to become minimax approximations. It was pointed out that discrete approximation was more practical, therefore from now on we restrict the discussion to discrete least pth approximation only.

We consider the minimization of the $\ell_p(n)$ norm (4.4) which is our objective

$$U(\underset{\sim}{a}) = \left(\sum_{i \in I} |e_i(\underset{\sim}{a})|^p\right)^{\frac{1}{p}} \quad \text{for} \quad 1 < p < \infty. \tag{5.1}$$

The scaling of functions and variables in an optimization problem is often crucial to the successful functioning of existing methods. Intuitively, we mean by "well-scaled" that similar changes in the function or variables lead to similar changes in the objective function. In a contour diagram of a well-scaled function of two variables the contour lines would not deviate too far from concentric circles and in this case we would hope that a method of steepest descent [12] would work satisfactorily. When the contours are exactly concentric circles, steepest descent gives the optimum in one step.

To alleviate the ill-conditioning in (5.1) for very large values of p, a scaling was proposed [1] which modifies all the terms which have to be raised to the power p such that they are less than or equal to unity. Let

$$M(\underset{\sim}{a}) \overset{\Delta}{=} \max_{i\in I} |e_i(\underset{\sim}{a})| \tag{5.2}$$

and rewrite (5.1) in the form

$$U(\underset{\sim}{a}) = M(\underset{\sim}{a})\left(\sum_{i\in I} \left|\frac{e_i(\underset{\sim}{a})}{M(\underset{\sim}{a})}\right|^p\right)^{\frac{1}{p}} . \tag{5.3}$$

It is always true for all $i\in I$ that

$$\left|\frac{e_i(\underset{\sim}{a})}{M(\underset{\sim}{a})}\right| \leq 1 \tag{5.4}$$

where there is at least one i for which equality holds. Large errors will be emphasized by large values of p and the contribution of the other terms in the summation will become very small.

Assuming that the $e_i(\underset{\sim}{a})$ for all $i\in I$ are continuous with continuous derivatives, the gradient of (5.3) is given by

$$\underset{\sim}{\nabla} U(\underset{\sim}{a}) = \left(\sum_{i\in I} \left|\frac{e_i(\underset{\sim}{a})}{M(\underset{\sim}{a})}\right|^p\right)^{\frac{1-p}{p}} \sum_{i\in I} \left|\frac{e_i(\underset{\sim}{a})}{M(\underset{\sim}{a})}\right|^{p-2} \cdot \mathrm{Re}\left(\frac{e_i^*(\underset{\sim}{a})}{M(\underset{\sim}{a})} \underset{\sim}{\nabla} e_i(\underset{\sim}{a})\right) \tag{5.5}$$

where

$$\underset{\sim}{\nabla} \triangleq \begin{bmatrix} \frac{\partial}{\partial a_1} \\ \frac{\partial}{\partial a_2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial}{\partial a_k} \end{bmatrix} \tag{5.6}$$

and * denotes the complex conjugate.

We note that $M(\underset{\sim}{a})$ and also $U(\underset{\sim}{a})$ is positive and $1<p<\infty$.

5.2 Quadratic interpolation for a function of a single variable

It is assumed that a sufficient number of sample points have been chosen along the x axis so that the discrete approximation problem adequately approximates the continuous problem. However, it should be remembered that function evaluations are often by far the most time consuming parts of an optimization process. So the number of sample points should be carefully chosen for the particular problem under consideration. These arguments apply, of course, to any formulation which involves sampling. If the requirement is to concentrate more on minimizing the maximum deviation it is suitable to sample points in the neighbourhood of the maxima of the weighted error function. As one usually cannot know the positions of the maxima in advance, it is common to space the sample points uniformly along the axis. Retaining the

maxima among them and reducing the number of the sample points by removing from the objective function those which do not contribute to the summation considerably may save computation time. Even more can be done if the approximations to the actual maxima replace the sample points in their neighborhood.

It is assumed that, in the neighborhood of the extremum, the function is adequately represented by a quadratic form. This assumption is the same as the ones we made in Chapter III, where we assume that in the neighborhood of the minimum the function to be optimized can be approximated by a quadratic form. The function is evaluated at three points, a quadratic interpolation polynomial is fitted to it, and the maximum of this interpolant is formed. This point replaces one of the initial points. This assumption has been thoroughly tested in practice and found satisfactory.

The basic algorithm involving quadratic interpolation applied to the weighted error function which is a multimodal function may be explained as follows:

Assume that the weighted error function is continuous on the closed interval $[x_a, x_b]$. Let

$$\{x_i^{(j)}\} \quad , \quad i \in I \tag{5.7}$$

be the set of n sample points at the beginning of the jth iteration. Before the searching procedure for the extrema of the weighted error starts, the end points of the interval are added to the given set (5.7),

if they are not already included. A set $\{x_i^{(j)}\}$ is used to construct $n+1$ (or $n+3$, if $x_1 \neq x_a$ and $x_n \neq x_b$) subintervals over $[x_a, x_b]$. Each subinterval is divided by a predicted n_s equidistant grid of points. Let

$$\{z_k^{(i,j)}\}\ ,\quad k=1,2,\ldots,n_s+1 \tag{5.8}$$

be the set of equidistant grid of points on $[x_i^{(j)}, x_{i+1}^{(j)}]$ interval. The extrema of the error function are found by the sequential examination of the values of the weighted error function and by comparison with the greatest on the subinterval obtained up to that time. If it happened that both neighbouring points have absolute values of the weighted error less than the current one, then the extremum $\hat{z}_i^{(i,j)}$ is found by applying the quadratic interpolation

$$\hat{z}_k^{(i,j)} = z_k^{(i,j)} + \frac{|e_{k+1}|-|e_{k-1}|}{2|e_k| - |e_{k+1}| - |e_{k-1}|} \cdot \Delta_i \tag{5.9}$$

where

$$e_k \overset{\Delta}{=} e_k(z_k^{(i,j)}) \tag{5.10}$$

and

$$\Delta_i \overset{\Delta}{=} z_{k+1}^{(i,j)} - z_k^{(i,j)} = \frac{x_{i+1}^{(j)} - x_i^{(j)}}{n_s}\ . \tag{5.11}$$

This point replaces $x_i^{(j)}$, the left end point of the current subinterval.

Immediately after the extreme point is located on the grid, the point on the grid next to the extremum replaces the left end point of the next subinterval. This is done in case there are more than one

extremum on a single subinterval. Thus the other extrema, if they exist, are "removed" to the next subinterval by removing its left end point. However, if it is found that there are no extreme points on the next subinterval, the end point is again set to its previous value.

All the extrema are selected by applying this searching procedure on every subinterval and these points replace the nearby left hand side point obtained up to that time, and the new set $\{x_i^{(j+1)}\}$, $i \in I$ for the (j+1)th iteration is thus obtained. This set does not necessarily contain the end points of the interval, but they are included in the searching technique at the next iteration to avoid shrinking the interval.

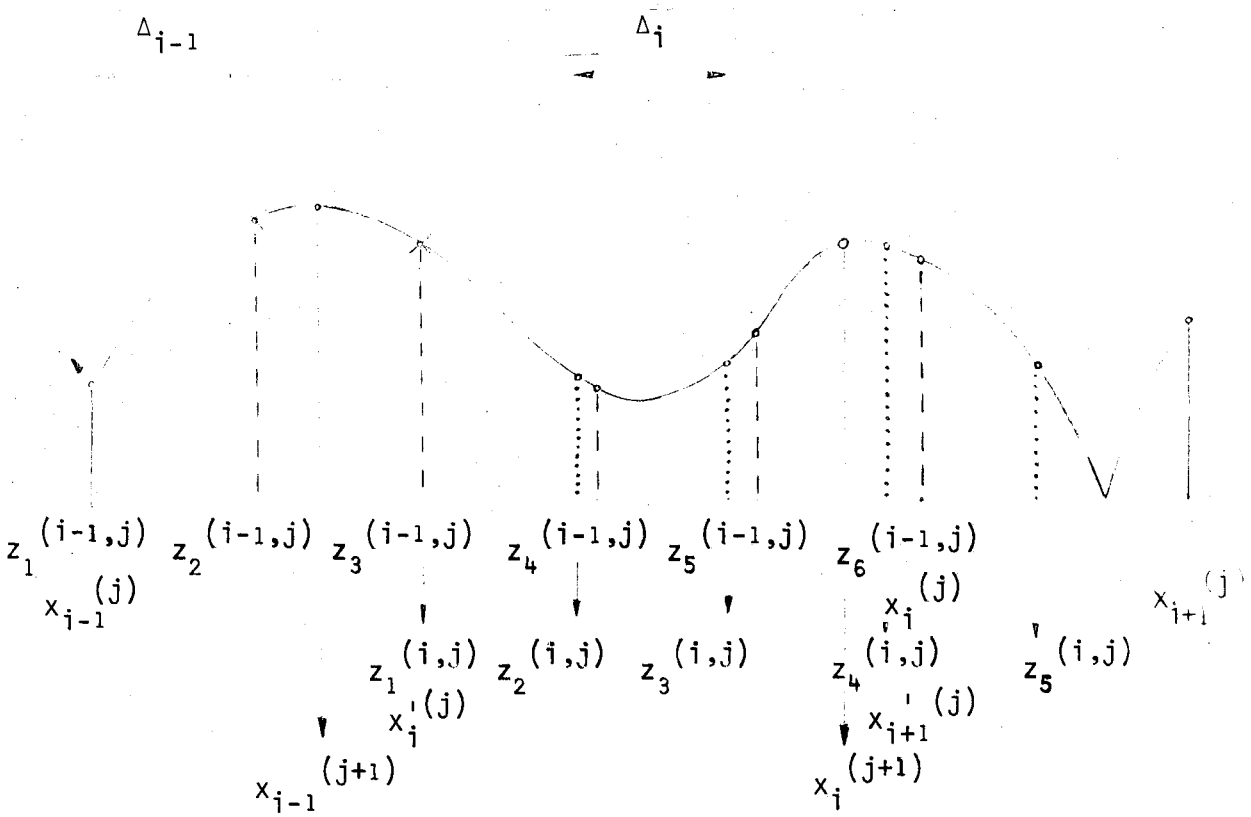


Fig. 5-1. Iterative searching procedure for finding maxima on two subintervals.

In an effort to keep the number of the discrete points in the summation (5.3) constant, at the end of the searching techniques we select n sample points among n+2 according to the absolute value of the corresponding weighted error function. If the error at one end point of the interval has a considerably large value relative to the other, the other is omitted. But if both end points have considerable errors, two neighbouring points are omitted among four successive internal points where the absolute values of the weighted error function are the smallest.

A scheme of this procedure for $n_s = 5$ is illustrated in Fig. 5-1.

Selection of the extreme points is significant especially within the first iteration. Once they are found, they do not usually move too far away in the next iterations.

The following is a list of some arguments used for the quadratic interpolation technique in the subroutine NEWSET:

indic(1) may have values 1 or 2.

indic(1)=1 indicates that quadratic interpolation is not applied on a subinterval;

indic(1)=2 indicates that a quadratic interpolation is done, and the left end point of the next subinterval is temporarily removed.

indic(2) may have values 1 or 2.

indic(2)=1 indicates that quadratic interpolation was not

applied on a previous subinterval;

indic(2)=2 indicates that a left end point of the subinterval was temporarily removed and if indic(1)=2, the left end point will be set to the new extreme point on the subinterval, otherwise it will be fixed to the value it had before the searching technique was applied.

indic(3) may have values 2, 1, -1 and -2.

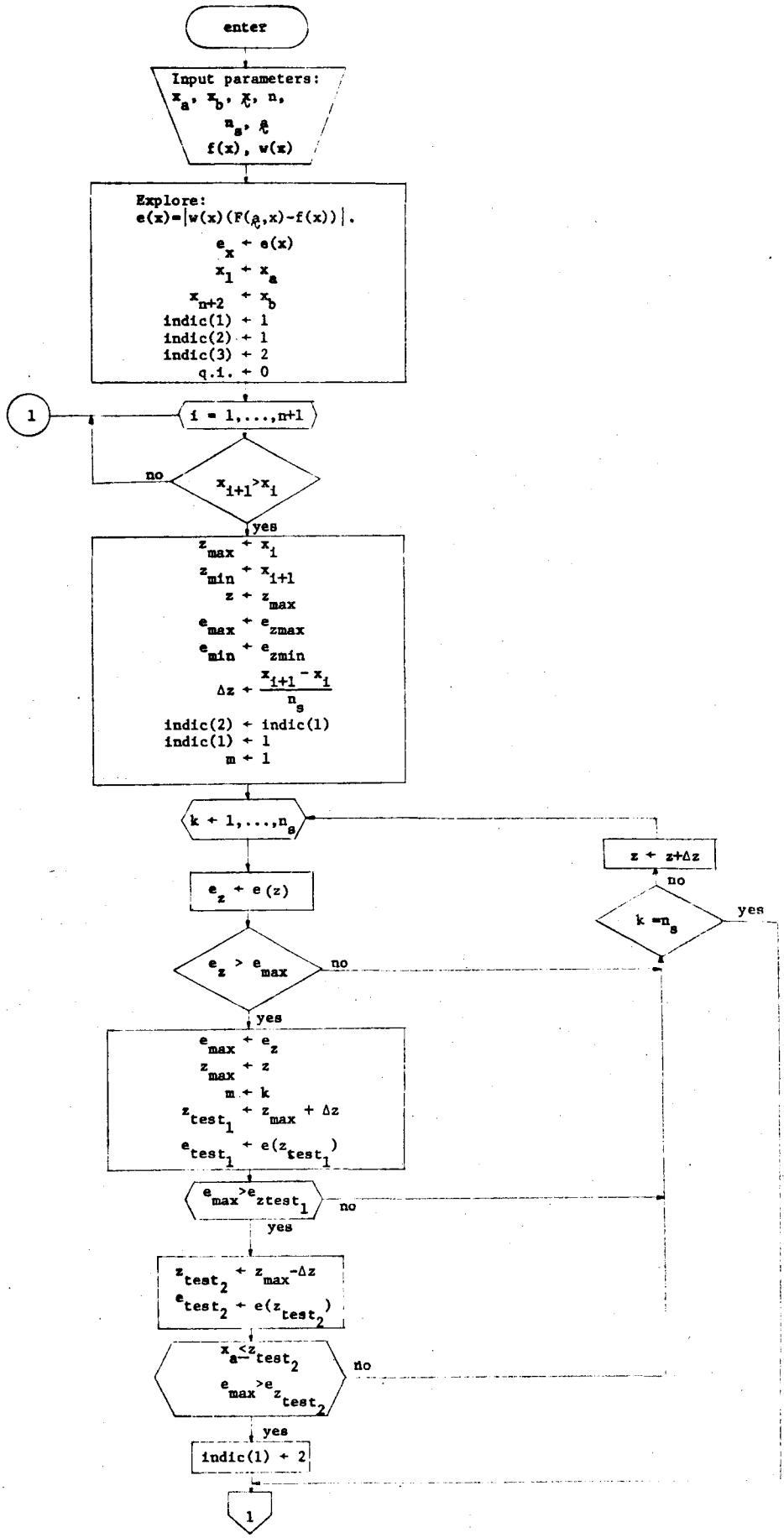
indic(3)=2 indicates that a new set of points does not include the left and the right end points of the interval $[x_a,x_b]$;

indic(3)=1 indicates that x_b is included in the new set of points;

indic(3)=-1 indicates that x_a is included in the new set of points;

indic(3)=-2 indicates that both x_a and x_b are included in the new set of points.

q.i. indicates the number of quadratic interpolations done on the interval $[x_a,x_b]$.


enter
Input parameters:
x_a, x_b, $\underset{\sim}{x}$, n,
n_s, $\underset{\sim}{a}$
f(x), w(x)
Explore:
$e(x)=|w(x)(F(\underset{\sim}{a},x)-f(x))|$.
$e_x \leftarrow e(x)$
$x_1 \leftarrow x_a$
$x_{n+2} \leftarrow x_b$
indic(1) ← 1
indic(2) ← 1
indic(3) ← 2
q.i. ← 0
1
i = 1,...,n+1
no
$x_{i+1} > x_i$
yes
$z_{max} \leftarrow x_i$
$z_{min} \leftarrow x_{i+1}$
$z \leftarrow z_{max}$
$e_{max} \leftarrow e_{zmax}$
$e_{min} \leftarrow e_{zmin}$
$\Delta z \leftarrow \frac{x_{i+1} - x_i}{n_s}$
indic(2) ← indic(1)
indic(1) ← 1
m ← 1
k ← 1,...,n_s
$e_z \leftarrow e(z)$
$e_z > e_{max}$
no
yes
$e_{max} \leftarrow e_z$
$z_{max} \leftarrow z$
m ← k
$z_{test_1} \leftarrow z_{max} + \Delta z$
$e_{test_1} \leftarrow e(z_{test_1})$
$e_{max} > e_{ztest_1}$
no
yes
$z_{test_2} \leftarrow z_{max} - \Delta z$
$e_{test_2} \leftarrow e(z_{test_2})$
$x_a < z_{test_2}$
$e_{max} > e_{z_{test_2}}$
no
yes
indic(1) ← 2
1
$z \leftarrow z + \Delta z$
no
$k = n_s$
yes

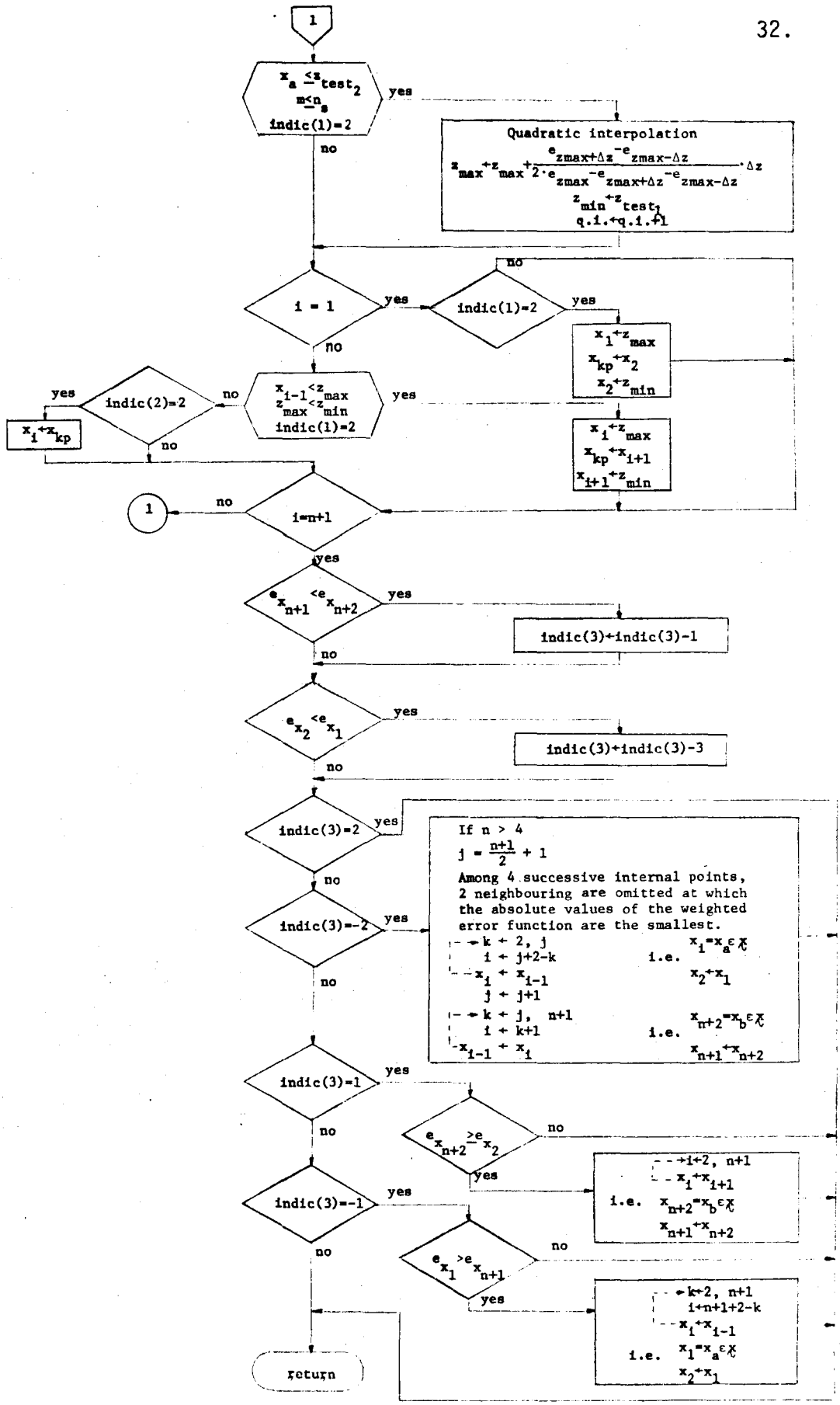


Fig. 5-2 Flowchart of subroutine NEWSET

5.3 Computer program

A computer program in FORTRAN IV is written for minimizing a function defined as a least pth objective where all the functions from (4.1) are real. Although the program is used simply by calling the name FMCLP, it contains 14 different subprograms. A list and a brief description of all the subprograms is given below:

FMCLP	Supplies all the data for the function minimization and coordinates the other subprograms in the package;
S	Defines a specified function over an interval;
FAPP	Defines an approximating function over an interval and the gradients w.r.t. variable parameters;
W	Defines a weighting function over an interval;
WERR	Computes a weighted error function at each sample point;
NEWSET	Redefines a sample point set using quadratic interpolation to include all the extreme points;
FUNCT	Finds the absolute maximum weighted error and computes the least pth objective (5.3) and its gradients (5.5);
GRDCHK	Checks the gradients w.r.t. all variable parameters

before the optimization process starts;

FMFPC — Minimizes a function using the Fletcher-Powell optimization technique described in the section 3.3;

FMNFC — Minimizes a function using the Fletcher optimization technique described in the section 3.4;

INPUT — Prints the input data for the optimization process;

FINAL — Prints the optimum solution after function minimization;

WRITE1 and WRITE2 — Print the intermediate results.

S, W and WERR are function subprograms and the others are subroutine subprograms.

A user of the package FMCLP is supposed to write the following subprograms from the above list: S, FAPP and W where he defines his own design problem in a straightforward way. More detailed instructions about how this should be done to fit the given package are available in Appendix I.

Subroutine NEWSET is written according to the algorithm and the flowchart given in Section 5.2. The other subprograms require more explanation.

The output of the function subprogram WERR has a value of the weighted error at a single point x for a particular vector $\underset{\sim}{a}$, i.e.

$$\text{WERR} \overset{\Delta}{=} (F(\underset{\sim}{a},x)-S(x))\cdot w(x) \quad . \tag{5.12}$$

This subprogram is called for each sample point as many times as there are the changes in the vector $\underset{\sim}{a}$.

Subroutine FUNCT keeps the values of the weighted error of each sample point in an n-dimensional array, finds the maximum absolute value among them and computes the objective function and its gradients according to the equations (5.3) and (5.5), respectively. The flow-chart of FUNCT is given in Fig. 5-4.

Subroutine GRDCHK checks the gradients of the approximating function, which a user has to supply, indirectly through the gradients of the objective function (5.5), before the optimization procedure starts. The criterion used for checking the gradient is the relative comparison between the analytical and the numerical gradients against a small given number η

$$\frac{\dfrac{U(a_i+\Delta a_i)-U(a_i)}{\Delta a_i}-\dfrac{\partial U}{\partial a_i}}{\dfrac{U(a_i+\Delta a_i)-U(a_i)}{\Delta a_i}} < \eta \quad , \quad i=1,\ldots,k \tag{5.13}$$

where in particular $\Delta=10^{-4}$ and $\eta=10^{-1}$. If it happens that

$$\frac{U(a_i+\Delta a_i)-U(a_i)}{\Delta a_i}<10^{-20} \quad , \quad i=1,\ldots,k \ , \tag{5.14}$$

then a small quantity 10^{-20} replaces the denominator in (5.13) to avoid the possible division by zero in that case. The gradients are checked only once, for the starting values of $\underset{\sim}{x}$ and $\underset{\sim}{a}$. If (5.13) does not hold, the program will terminate and wait for a user's intervention.

FMFPC and FMNFC are subroutines available for function minimization. The user has to specify which method is going to be used, how many times and the stopping criteria for the minimization procedure. There is the possibility of choosing both at one run. For more detailed instructions see Appendix I.

The subroutines INPUT, FINAL, WRITE1 and WRITE2 are called only when certain data or results are desired to be printed out. How this is handled is explained in Appendix I.

All the subprograms are coordinated by FMCLP. Fig. 5-3 illustrates how all these subprograms are tied to FMCLP and is useful in understanding the organization of the whole program. A flowchart of FMCLP, given in Fig. 5-5, shows in more details how this is actually done.

The program terminates when the stopping criteria for the optimization method, Fletcher-Powell or Fletcher method, are satisfied or when the relative change in the objective function in two successive iterations is less than a small prescribed quantity ε, i.e.,

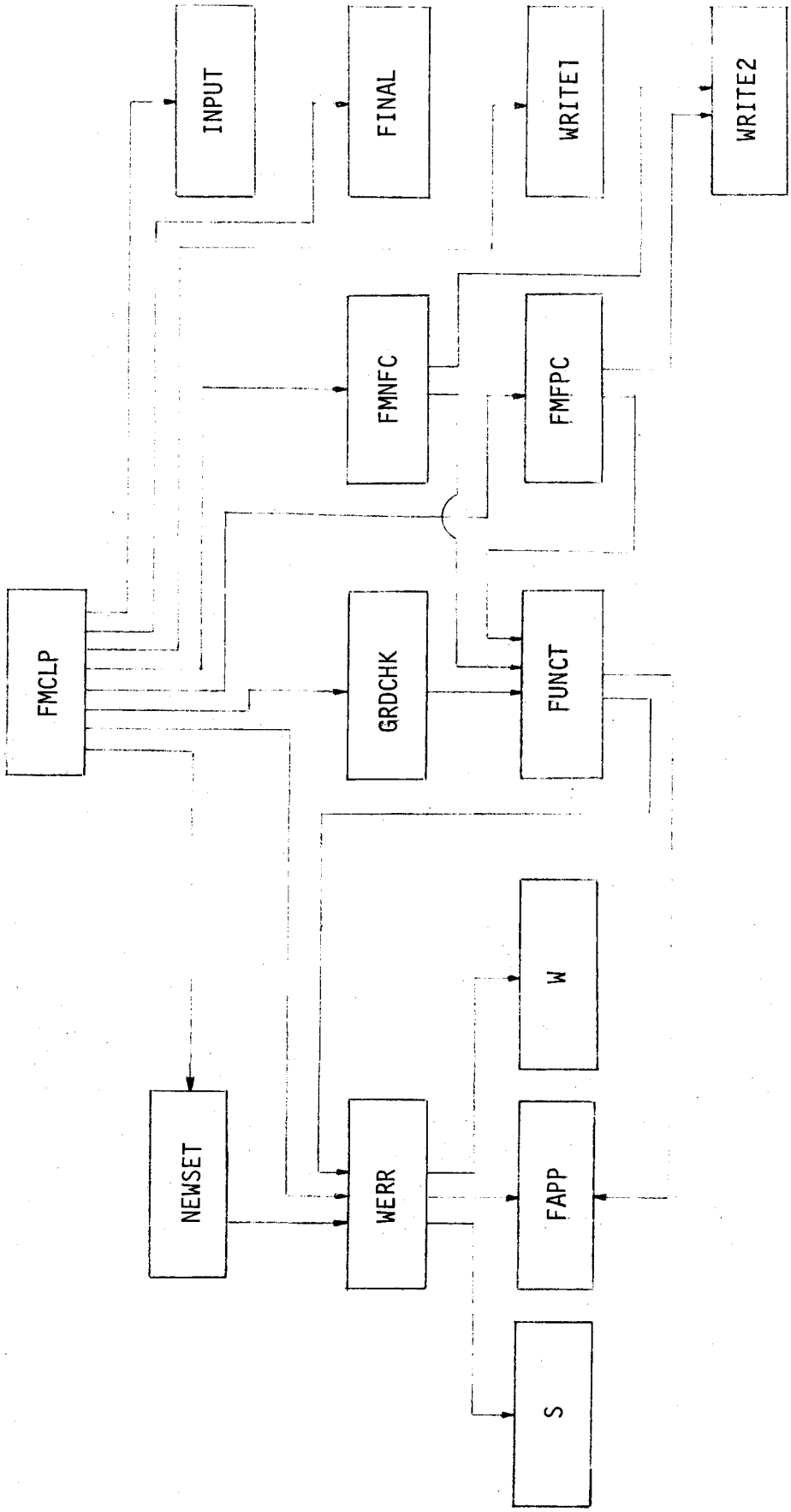


Fig. 5-3 The organization of FMCLP.

$$\frac{U(\underset{\sim}{a}^{j-1})-U(\underset{\sim}{a}^{j})}{U(\underset{\sim}{a}^{j-1})} < \varepsilon \quad . \tag{5.15}$$

The listing of the program described in this Chapter is attached in Appendices II and III.

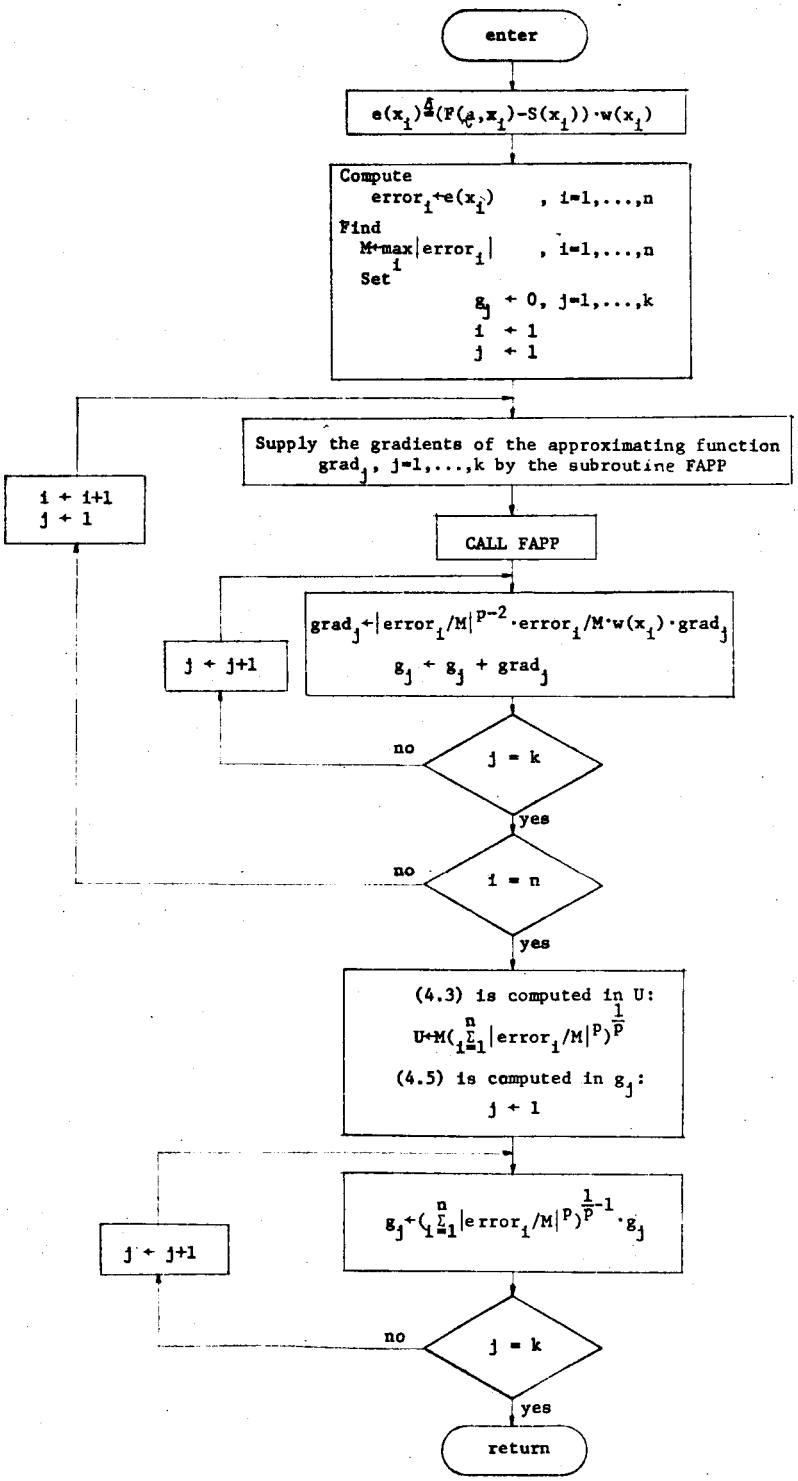
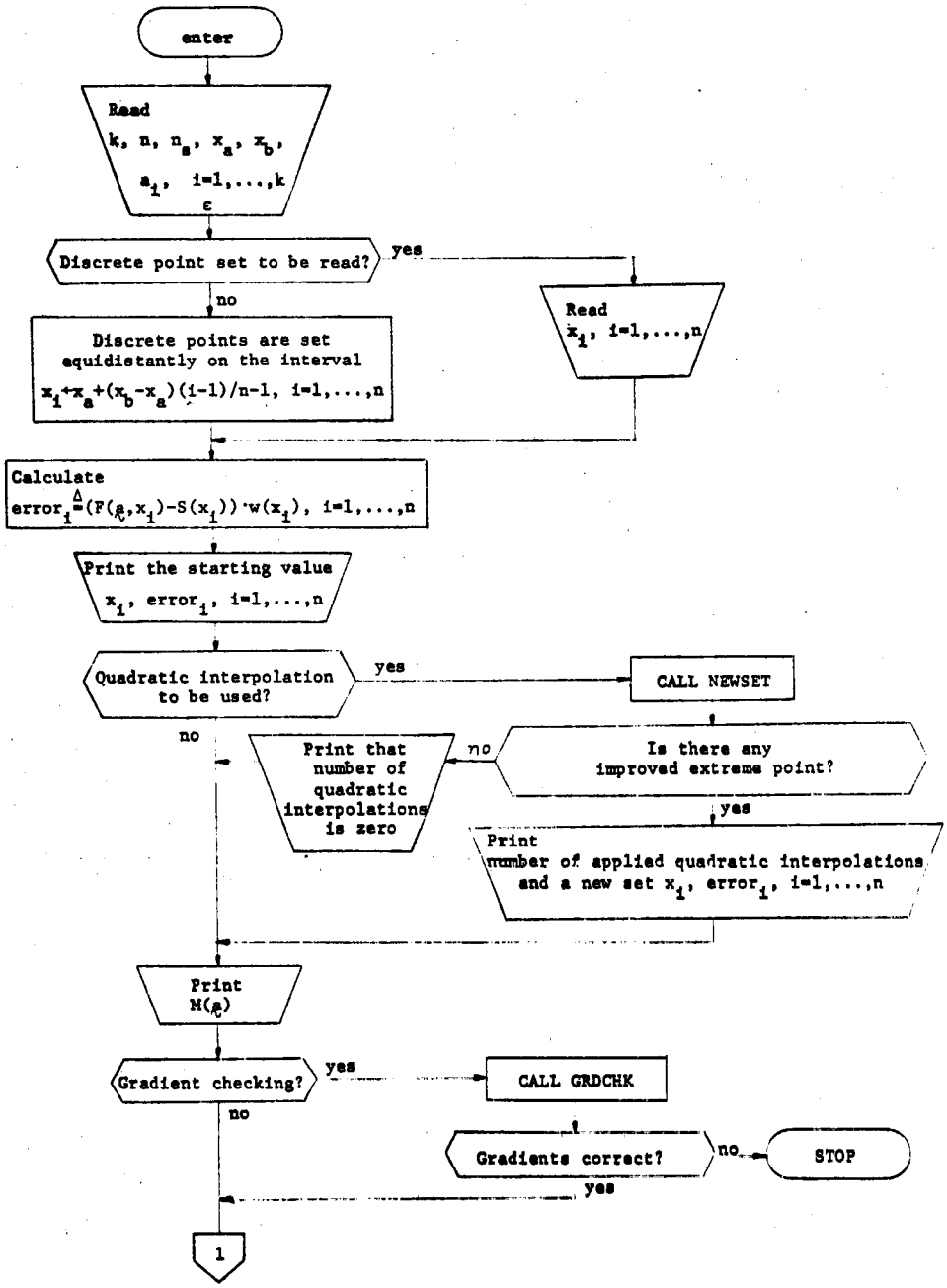


Fig. 5-4 Flowchart of subroutine FUNCT


enter
Read
k, n, n_s, x_a, x_b,
a_i, i=1,...,k
ε
Discrete point set to be read?
yes
no
Read
x_i, i=1,...,n
Discrete points are set equidistantly on the interval
$x_i \leftarrow x_a + (x_b - x_a)(i-1)/n-1$, i=1,...,n
Calculate
$error_i \triangleq (F(\underline{a}, x_i) - S(x_i)) \cdot w(x_i)$, i=1,...,n
Print the starting value
x_i, $error_i$, i=1,...,n
Quadratic interpolation to be used?
yes
CALL NEWSET
no
Print that number of quadratic interpolations is zero
no
Is there any improved extreme point?
yes
Print
number of applied quadratic interpolations and a new set x_i, $error_i$, i=1,...,n
Print
$M(\underline{a})$
Gradient checking?
yes
CALL GRDCHK
no
Gradients correct?
no
STOP
yes
1

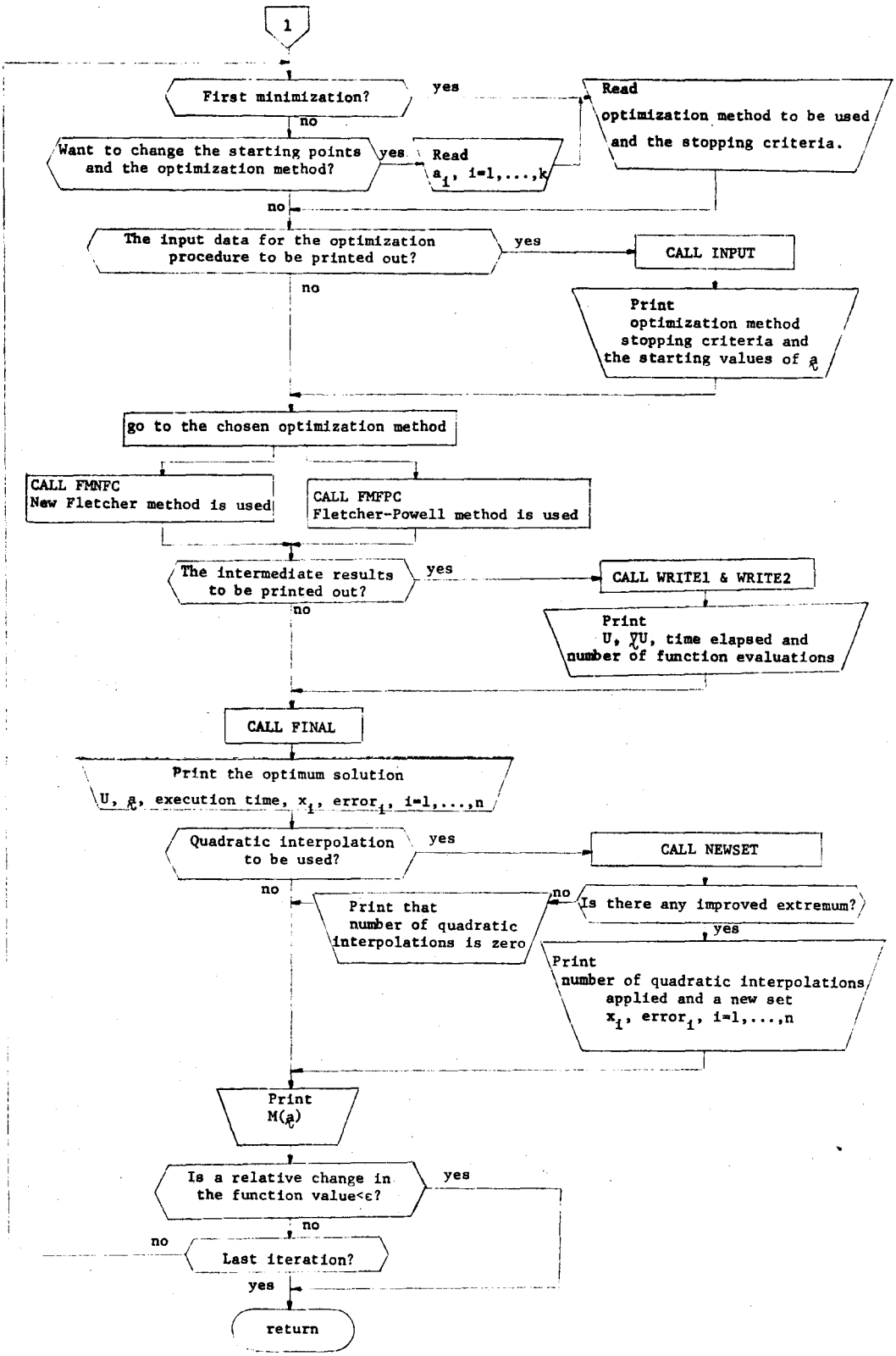


Fig. 5-5 Flowchart of subroutine FMCLP

5.4 Examples

As mentioned before, using larger values of p, the more nearly is the maximum error emphasized and with extremely large values of p we approach the minimax solution. Two numerical examples, where the rational minimax approximations to the functions are known, were taken for comparison with results obtained using least pth approximation [22]. The reason why rational approximating functions are chosen in the least pth objective is that adequate comparison between minimax and least pth approximation may be done, and there exists a unique rational function

$$R_{\ell m}(x) = \frac{P_\ell(x)}{Q_m(x)} = \frac{\sum_{j=0}^{\ell} a_j x^j}{1 + \sum_{j=1}^{m} b_j x^j} \tag{5.16}$$

which minimizes the uniform norm. The uniqueness is proved by Chebyshev [30]. In this case the defect has to be included in Chebyshev's theorem. It is known that if the functions form a Chebyshev set, the best approximation to a function over a closed interval is characterized by the maximum error occurring at k+1 points, the sign of error alternating.

The initial approximation and the starting point set in the examples were chosen to be the same as in the minimax approximation problem [22]. The initial approximation was obtained by rational

interpolation where the zeros of the Chebyshev polynomial (k+1)st order, T_{k+1} [3]], transformed on the interval $[x_a, x_b]$ are used as supporting points. As a first trial for the point set $\underset{\sim}{x}$, the extrema of the (n+1)st order Chebyshev polynomial were taken. There is no special reason for choosing these initial points with least pth objective, but it is shown that they provide a good initial guess for the minimax algorithm. A computer program which calculates the starting values of $\underset{\sim}{a}$ and $\underset{\sim}{x}$ is given in Appendix IV.

Test 1

Approximate

$$f(x) = e^x - 1.1 \tag{5.17}$$

on the interval [-1,1] by a rational function (5.16) with $\ell=1$ and $m=3$. Use the weighting function $w(x)=1$.

An initial set of 12 points is chosen over an interval [-1,1] and all the test quantities for the Fletcher-Powell and Fletcher method were set to 10^{-6}. The value of $p=10^4$ was found to be large enough to bring the least pth close to the minimax solution. The results are presented in Tables 5-1 and 5-2.

From Table 5-1 we may draw the conclusion that good results were obtained with both optimization techniques. Also, it may be noticed that the Fletcher method was more efficient than the

	Starting values	Minimax solution [22]	Least pth solution for $p=10^4$	
			Fletcher-Powell	Fletcher
a_0	-1.00000×10^{-1}	-9.99928×10^{-2}	-9.99274×10^{-2}	-9.99274×10^{-2}
a_1	1.04878	1.04887	1.04887	1.04887
b_1	-4.92007×10^{-1}	-4.91425×10^{-1}	-4.91425×10^{-1}	-4.91425×10^{-1}
b_2	7.77810×10^{-2}	7.77872×10^{-2}	7.78725×10^{-2}	7.78725×10^{-2}
b_3	6.04748×10^{-4}	-2.60009×10^{-5}	-2.59776×10^{-5}	-2.60049×10^{-5}
U or M		$M=8.68978\times10^{-5}$	$U=8.69170\times10^{-5}$	$U=8.69326\times10^{-5}$
Number of iterations		4	5	5
Number of function evaluations		-	420	115
Execution time in seconds		-	14	3.5

Table 5-1

Minimax solution		Least pth solution for $p=10^4$			
		Fletcher-Powell		Fletcher	
$\underset{\sim}{x}$ where the extreme errors occur	extreme errors	working set of $\underset{\sim}{x}$ at the optimum	error ($\underset{\sim}{a}$) at the optimum	working set of $\underset{\sim}{x}$ at the optimum	error ($\underset{\sim}{a}$) at the optimum
-1.00000	8.68978×10^{-5}	-1.00000	8.68987×10^{-5}	-1.00000	8.69123×10^{-5}
		-9.59493×10^{-1}	3.42715×10^{-5}	-9.59493×10^{-1}	3.42850×10^{-5}
-7.23512×10^{-1}	-8.68938×10^{-5}	-7.25986×10^{-1}	-8.68983×10^{-5}	-7.25929×10^{-1}	-8.68834×10^{-5}
		-7.25825×10^{-1}	-8.68983×10^{-5}	-7.25829×10^{-1}	-8.68833×10^{-5}
-1.19232×10^{-1}	8.68977×10^{-5}	-1.19127×10^{-1}	-8.68976×10^{-5}	-1.19111×10^{-1}	8.69252×10^{-5}
		-1.18506×10^{-1}	8.68972×10^{-5}	-1.18495×10^{-1}	8.69248×10^{-5}
4.72848×10^{-1}	-8.68975×10^{-5}	4.73739×10^{-1}	-8.69024×10^{-5}	4.73912×10^{-1}	-8.68853×10^{-5}
		4.82411×10^{-1}	-8.67694×10^{-5}	4.82510×10^{-1}	-8.67500×10^{-5}
8.65372×10^{-1}	8.68972×10^{-5}	8.65710×10^{-1}	8.69016×10^{-5}	8.65783×10^{-1}	8.68912×10^{-5}
		8.99065×10^{-1}	7.93572×10^{-5}	8.99070×10^{-1}	7.93431×10^{-5}
		9.12197×10^{-1}	7.15451×10^{-5}	9.12197×10^{-1}	7.15331×10^{-5}
1.00000	-8.68977×10^{-5}	1.00000	-8.68953×10^{-5}	1.00000	-8.69069×10^{-5}

Table 5-2

Fletcher-Powell method because it requires fewer function evaluations and is less time-consuming.

Test 2

Approximate

$$f(x) = \frac{\sqrt{(8x-1)^2 + 1}\cdot\tan^{-1}(8x)}{8x} \tag{5.18}$$

with respect to the weighting function $w(x)=1$ on the interval $[-1,1]$ by a rational function with $\ell=2$ and $m=2$.

This example is remarkable because it works near degeneracy defined in the Chebyshev theorem [30], and meets the artificial poles for almost every combination of ℓ and m in the rational function (5.16) [21]. In true degeneracy of degree one, there is one less extremum of the error curve than is expected with a minimax algorithm, and essentially the numerator and denominator of the rational approximation contain a common linear factor which must be divided out to reduce the rational function to its lowest terms. In the numerical examples it is not at all unusual to find that the numerator has one zero that differs from a denominator zero by some small amount, i.e., to have a rational approximation nearly degenerate of degree one.

The comparative results between the minimax and least pth approximation obtained by the Fletcher optimization technique for $p=10^4$ are presented in Tables 5-3 and 5-4.

	Starting values	Minimax solution	Least pth solution for $p=10^4$ obtained by the Fletcher method
a_0	1.00000×10^{-2}	1.41450	1.41448
a_1	-3.33600	-1.06530×10^{1}	-1.06519×10^{1}
a_2	4.76782×10^{1}	4.16169×10^{1}	4.16157×10^{1}
b_1	1.76567	-4.0103	-4.00940
b_2	3.19620×10^{1}	2.82628×10^{1}	2.82620×10^{1}
U or M		$M=2.38113 \times 10^{-2}$	$U=2.38154 \times 10^{-2}$
Number of iterations		8	13
Number of function evaluations		-	586
Execution time in seconds		-	9.2

Table 5-3

Minimax solution		Least pth solution for $p=10^4$ obtained by the Fletcher method	
x where the extreme errors occur	extreme errors	working set of x at the optimum	errors at the optimum
-1.00000	-2.38108×10^{-2}	-1.00000	-2.38025×10^{-2}
		-9.59493×10^{-1}	-2.15823×10^{-2}
		-4.34766×10^{-1}	1.77356×10^{-2}
-3.08573×10^{-1}	2.38108×10^{-2}	-3.15535×10^{-1}	2.38026×10^{-2}
-6.15510×10^{-2}	-2.38109×10^{-2}	-6.18947×10^{-2}	-2.38007×10^{-2}
5.42007×10^{-2}	2.38108×10^{-2}	5.46342×10^{-2}	2.37994×10^{-2}
		1.93965×10^{-1}	-2.37976×10^{-2}
1.96670×10^{-1}	-2.38113×10^{-2}	1.94011×10^{-1}	-2.37976×10^{-2}
5.52892×10^{-1}	2.38108×10^{-2}	5.52292×10^{-1}	2.38004×10^{-2}
		6.20236×10^{-1}	2.27473×10^{-2}
		8.09651×10^{-1}	1.42303×10^{-2}
		1.00000	3.68693×10^{-3}

Table 5-4

Test 3

There was an attempt by Rice to extend the Chebyshev theorem to include non-linear dependence of $F(\underset{\sim}{a},x)$ on $\underset{\sim}{a}$ that the same characterization of best approximations holds for all continuous $S(x)$ [32], but this applies only to a limited class of approximating functions [20]. Unfortunately, there are many useful choices of $F(\underset{\sim}{a},x)$ which do not fulfil Rice's conditions of best approximations [33].

The third example is another that might be expected to give trouble. Due to Curtis and Powell [33], it is the approximation of x^2 by $a_1x+a_2e^x$ over $0 \leq x \leq 2$. It may be verified that the error function of the approximation

$$x^2 \approx 8.465x - 2.0239e^x \tag{5.19}$$

takes its maximum absolute value at the three points x=0, x=1.1227 and x=2, the error at these points being +2.0239, -2.0239 and +2.0239, respectively. In fact, the best approximation is

$$x^2 \approx 0.1842x + 0.4186e^x \tag{5.20}$$

the maximum absolute error is 0.5382 and this error occurs at just the two points: x=0.4064 and x=2. Not only do the approximating functions fail to form a Chebyshev set, but also the error curve has only two extrema instead of the three that would normally be anticipated according to the Rice's theorem. The least pth results for this problem are given in Table 5-5 and again show the success of

FMCLP. The estimates of the best approximation agree to four figures with those given by Curtis and Powell.

Initial approximation $\begin{bmatrix}1\\1\end{bmatrix}$	
Fletcher method:	
Stopping criteria 10^{-6}	
n	10
p	10^5
$\underset{\sim}{a}$	0.1848
	0.4184
$M(\underset{\sim}{a})$	0.5382
$\underset{\sim}{x}_M$	0.4066
	2.000
$U(\underset{\sim}{a})$	0.5382
function evaluation	67
execution time in seconds	1.3

Table 5-5

Test 4

The computer program is used to solve an electrical engineering design problem. The problem is stated: find a second order model of a fourth-order system with a given transfer function

$$G(s) = \frac{s+4}{(s+1)(s^2+4s+8)(s+5)} \quad . \tag{5.21}$$

The transfer function of the second-order model considered is

$$H(s) = \frac{a_3}{(s+a_1)^2 + a_2^2} \quad . \tag{5.22}$$

Using the inverse Laplace transform the responses for (5.21) and (5.22) are

$$S(t) = \frac{3}{20} e^{-t} + \frac{1}{52} e^{-5t} - \frac{1}{65} e^{-2t}(3\sin 2t + 11 \cos 2t) \tag{5.23}$$

and

$$F(\underset{\sim}{a},t) = \frac{a_3}{a_2} e^{-a_1 t} \cdot \sin a_2 t \quad , \tag{5.24}$$

respectively, where

$$\underset{\sim}{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad .$$

The results for different values of p and different numbers of sampling points n, but the same product $n \times n_s$, over the range $0 \leq t \leq 10$ are given in Table 5-6, agreeing with those given in [28].

Using the quadratic interpolation the locations of the extreme points were found precisely, and the solutions are closer to the minimax solution. Both solutions, for n=10 and n=25, are better than for n=50 where the quadratic interpolation was not employed. Moreover, the case n=10 is less time consuming, because this sampling takes the least number of points for the objective function (4.3) in comparison with the other two cases under consideration.

	$n=10,\ n_s=5$	$n=25,\ n_s=2$	$n=50,\ n_s=1$
p=2	$a_1=1.27339$ $a_2=6.54190\times10^{-1}$ $a_3=2.17787\times10^{-1}$ $M(\underset{\sim}{a})=2.05859\times10^{-2}$ $t_M=2.51460\times10^{-1}$ $U(\underset{\sim}{a})=4.71528\times10^{-3}$ f.e.=41 q.i.=2 1.6 sec	$a_1=1.05489$ $a_2=-7.67814\times10^{-1}$ $a_3=1.618192\times10^{-1}$ $M(\underset{\sim}{a})=1.32708\times10^{-2}$ $t_M=2.46234\times10^{-1}$ $U(\underset{\sim}{a})=1.21662\times10^{-2}$ f.e.=30 q.i.=3 2.3 sec	$a_1=1.01687$ $a_2=7.8915\times10^{-1}$ $a_3=1.61435\times10^{-1}$ $M(\underset{\sim}{a})=1.28696\times10^{-2}$ $t_M=2.04081\times10^{-1}$ $U(\underset{\sim}{a})=2.06679\times10^{-2}$ f.e.=36 q.i.=0 5.1 sec
p=10	$a_1=7.37873\times10^{-1}$ $a_2=9.2622\times10^{-1}$ $a_3=1.2862\times10^{-1}$ $M(\underset{\sim}{a})=8.9565\times10^{-3}$ $t_M=1.67727\times10^{-1}$ $U(\underset{\sim}{a})=8.4364\times10^{-3}$ f.e.=38 q.i.=3 1.5 sec	$a_1=7.46289\times10^{-1}$ $a_2=-9.23825\times10^{-1}$ $a_3=1.27596\times10^{-1}$ $M(\underset{\sim}{a})=8.76150\times10^{-3}$ $t_M=1.73062\times10^{-1}$ $U(\underset{\sim}{a})=8.2421\times10^{-3}$ f.e.=32 q.i.=3 2.5 sec	$a_1=7.43325\times10^{-1}$ $a_2=9.29377\times10^{-1}$ $a_3=1.2812\times10^{-1}$ $M(\underset{\sim}{a})=8.5446\times10^{-3}$ $t_M=2.04081\times10^{-1}$ $U(\underset{\sim}{a})=9.1834\times10^{-3}$ f.e.=31 q.i.=0 4.5 sec

	$n=10,\ n_s=5$	$n=25,\ n_s=2$	$n=50,\ n_s=1$
$p=10^2$	$a_1=6.79369\times10^{-1}$ $a_2=9.55429\times10^{-1}$ $a_3=1.21997\times10^{-1}$ $M(\underset{\sim}{a})=8.9563\times10^{-3}$ $t_M=1.67727\times10^{-1}$ $U(\underset{\sim}{a})=8.2055\times10^{-3}$ f.e.=35 q.i.=1 1.3 sec	$a_1=6.8047\times10^{-1}$ $a_2=-9.5471\times10^{-1}$ $a_3=1.2206\times10^{-1}$ $M(\underset{\sim}{a})=8.1300\times10^{-3}$ $t_M(\underset{\sim}{a})=1.73062\times10^{-1}$ $U(\underset{\sim}{a})=8.1866\times10^{-3}$ f.e.=35 q.i.=1 2.5 sec	$a_1=6.88905\times10^{-1}$ $a_2=9.52106\times10^{-1}$ $a_3=1.2334\times10^{-1}$ $M(\underset{\sim}{a})=7.9450\times10^{-3}$ $t_M=2.04081\times10^{-1}$ $U(\underset{\sim}{a})=8.0045\times10^{-3}$ f.e.=35 q.i.=0 5.sec
$p=10^3$	$a_1=6.737\times10^{-1}$ $a_2=9.5590\times10^{-1}$ $a_3=1.2168\times10^{-1}$ $M(\underset{\sim}{a})=8.1125\times10^{-3}$ $t_M=1.67727\times10^{-1}$ $U(\underset{\sim}{a})=8.1182\times10^{-3}$ f.e.=28 q.i.=0 1. sec	$a_1=6.77142\times10^{-1}$ $a_2=-9.5556\times10^{-1}$ $a_3=1.21735\times10^{-1}$ $M(\underset{\sim}{a})=8.0905\times10^{-3}$ $t_M=1.73062\times10^{-1}$ $U(\underset{\sim}{a})=8.0957\times10^{-3}$ f.e.=29 q.i.=0 2.3 sec	$a_1=6.8510\times10^{-1}$ $a_2=9.5289\times10^{-1}$ $a_3=1.2294\times10^{-1}$ $M(\underset{\sim}{a})=7.9009\times10^{-3}$ $t_M=2.04082\times10^{-1}$ $U(\underset{\sim}{a})=7.9068\times10^{-3}$ f.e.=26 q.i.=0 4. sec
Total	f.e.=142 for 5.4 sec	f.e.=126 for 9.6 sec	f.e.=128 for 18.6 sec

Table 5-6

Remark: The optimum result for a_2 is true for both positive and negative values from Table 5-6 because this does not effect the approximating function (5.24) where

$$\frac{\sin a_2 t}{a_2} \equiv \frac{\sin(-a_2 t)}{-a_2} \quad .$$

CHAPTER VI

Generalized Least pth Approximation

6.1 The objective functions

The applicability of least pth approximation may be extended to a wider variety of design problems and a wider range of specifications than considered in Chapter V if the least pth objective function is written in a more general form [2].

We will evaluate all the functions at a finite discrete set of values of x taken from one or more closed intervals.

Define real weighted error functions related to the upper and lower specifications, as shown in Fig. 6-1, respectively as

$$e_u(\underset{\sim}{a},x_i)\triangleq w_u(x_i)(F(\underset{\sim}{a},x_i)-S_u(x_i)), \quad i\in I_u$$

$$e_\ell(\underset{\sim}{a},x_i)=w_\ell(x_i)(F(\underset{\sim}{a},x_i)-S_\ell(x_i)) \ , \ i\in I_\ell \tag{6.1}$$

where

$F(\underset{\sim}{a},x_i)$	is the approximating function;
$Su(x_i)$	is an upper specified function;
$S_\ell(x_i)$	is an lower specified function;
$w_u(x_i)$	is an upper positive weighting function;
$w_\ell(x_i)$	is an lower positive weighting function;
$\underset{\sim}{a}$	is an vector containing the k independent parameters;

x_i is the discretized independent variable;

I_u and I_ℓ are appropriate index sets.

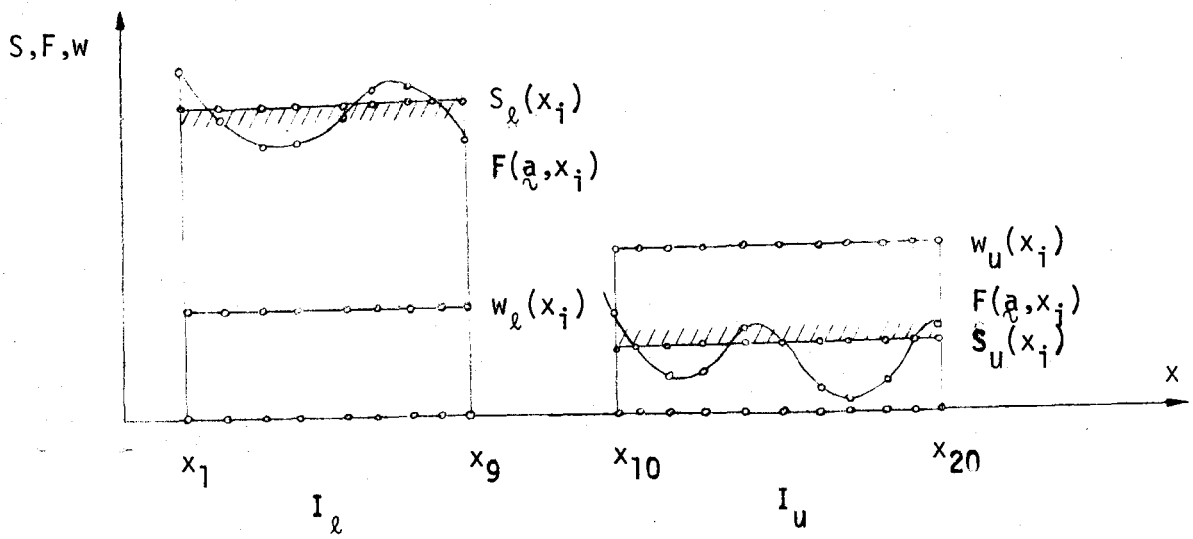


Fig. 6-1 Example of a design problem with upper and lower specifications.

A special case of (6.1) when $S_u=S_\ell$ and $w_u=w_\ell$ leads to the common form of a real weighted error function defined in (5.1).

We will slightly modify the error functions by introducing a constant ξ which is used for shifting the level of the originally defined errors functions (6.1) [34] such that

$$e_u'(\underset{\sim}{a},x_i,\xi)\triangleq e_u(\underset{\sim}{a},x_i)-\xi$$

(6.2)

$$e_\ell'(\underset{\sim}{a},x_i,\xi)\triangleq e_\ell(\underset{\sim}{a},x_i)+\xi \quad .$$

The above expressions may be obtained by redefining the

specifications which are going to be used in the error function (6.2) as follows

$$S_u'(x_i,\xi)=S_u(x_i)+\frac{\xi}{w_u(x_i)}$$

$$S_\ell'(x_i,\xi)=S_\ell(x_i)-\frac{\xi}{w_\ell(x_i)} \tag{6.3}$$

where $S_u'(x_i,\xi)$ and $S_\ell'(x_i,\xi)$ are the artificial upper and lower specified functions, respectively.

Therefore, the modified error functions are in the following forms

$$e_u'(\underset{\sim}{a},x_i,\xi)\overset{\Delta}{=}w_u(x_i)(F(\underset{\sim}{a},x_i)-S_u'(x_i,\xi))$$

$$e_\ell'(\underset{\sim}{a},x_i,\xi)\overset{\Delta}{=}w_\ell(x_i)(F(\underset{\sim}{a},x_i)-S_\ell'(x_i,\xi)). \tag{6.4}$$

A certain flexibility in the formulation of the optimization problem is possible when the artificial margin ξ is introduced. A more detailed discussion will come later.

Let

$$e_{u_i}(\underset{\sim}{a},\xi)\overset{\Delta}{=}e_u(\underset{\sim}{a},x_i,\xi)\ ,\quad i\in I_u$$

$$e_{\ell_i}(\underset{\sim}{a},\xi)\overset{\Delta}{=}e_\ell(\underset{\sim}{a},x_i,\xi)\ ,\quad i\in I_\ell\quad . \tag{6.5}$$

It is quite possible that some of the upper and lower error functions tend to $-\infty$ and $+\infty$, respectively, in which case they may be

simply ignored by a proper formulation of the problem.

We will consider two possible cases which may happen in function approximation with upper and lower specifications: the specification may be violated and the specification may be satisfied.

6.2 Case I-Specification Violated

When the specification is violated, some of the $e_{u_i}'(\underset{\sim}{a})$ or $-e_{\ell_i}'(\underset{\sim}{a})$ are positive, as is illustrated in Fig. 6-2. (The illustration is made for $\xi=0$, for convenience).

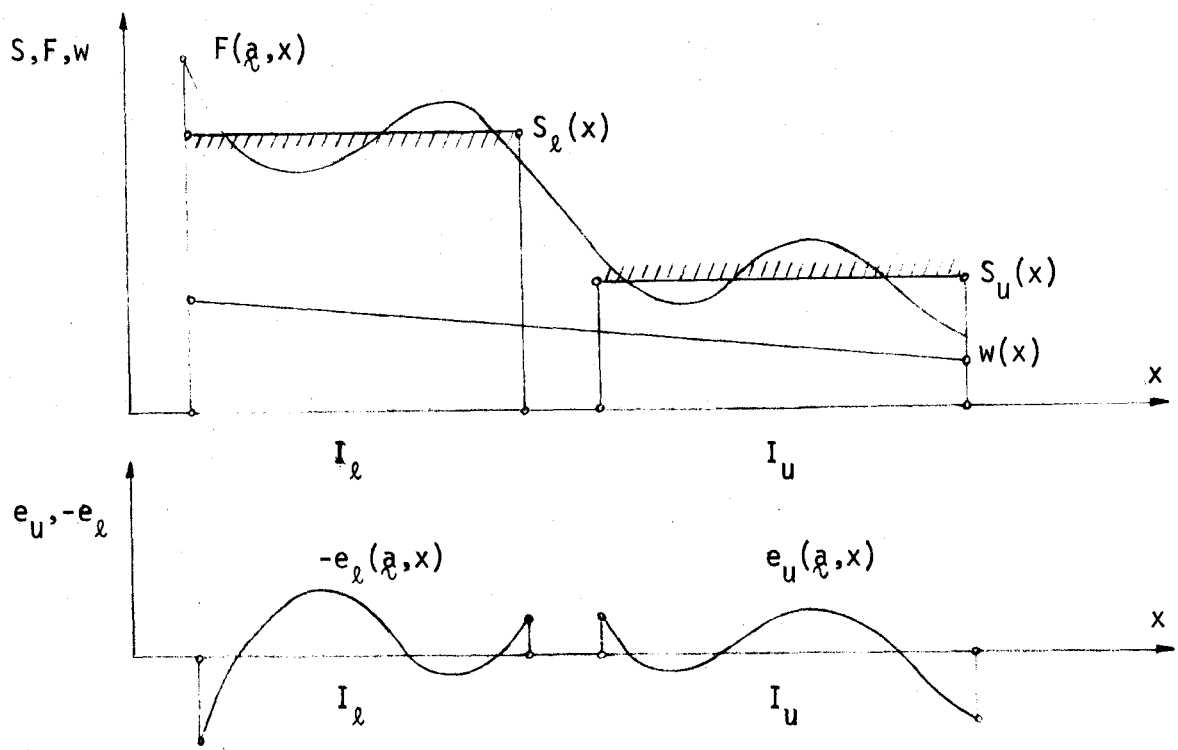


Fig. 6-2 Example of a design problem when the specification is violated. Case I is applicable.

We wish to make as small as possible all those $e_{u_i}(\underset{\sim}{a})$ and $-e_{\ell_i}(\underset{\sim}{a})$ which are positive, in an effort to meet the specification, and the following objective function was proposed [34] to be minimized:

$$U(\underset{\sim}{a},\xi)=\Big(\sum_{i\in J_u(\underset{\sim}{a},\xi)}[e_{u_i}'(\underset{\sim}{a},\xi)]^p+\sum_{i\in J_\ell(\underset{\sim}{a},\xi)}[-e_{\ell_i}'(\underset{\sim}{a},\xi)]^p\Big)^{\frac{1}{p}} \tag{6.6}$$

where

$$J_u(\underset{\sim}{a},\xi)\overset{\Delta}{=}\{i\,|\,e_{u_i}'(\underset{\sim}{a},\xi)\geq 0,\quad i\in I_u\}$$

$$J_\ell(\underset{\sim}{a},\xi)\overset{\Delta}{=}\{i\,|\,-e_{\ell_i}'(\underset{\sim}{a},\xi)\geq 0,\quad i\in I_\ell\} \tag{6.7}$$

and

$$p>1.$$

If J_u and J_ℓ are empty then $U(\underset{\sim}{a})=0$ and we have just met or exceeded the artificial specifications. The larger the value of p, the more nearly would we expect the maximum error to be emphasized, since

$$\max_{i,j}[e_{u_i}'(\underset{\sim}{a},\xi),\ -e_{\ell_j}'(\underset{\sim}{a},\xi)]=\lim_{p\to\infty}U(\underset{\sim}{a},\xi),\qquad \begin{matrix} i\in J_u(\underset{\sim}{a},\xi) \\ j\in J_\ell(\underset{\sim}{a},\xi)\end{matrix}, \tag{6.8}$$

hence minimization of (6.6) is an effort to meet the specification.

6.3 Case II-Specification is satisfied

When the specification is satisfied all the $-e_{u_i}'(\underset{\sim}{a},\xi)$ and $e_{\ell_i}'(\underset{\sim}{a},\xi)$ will be positive as it is shown in Fig. 6-3. Again for

convenience, $\xi=0$ is chosen,

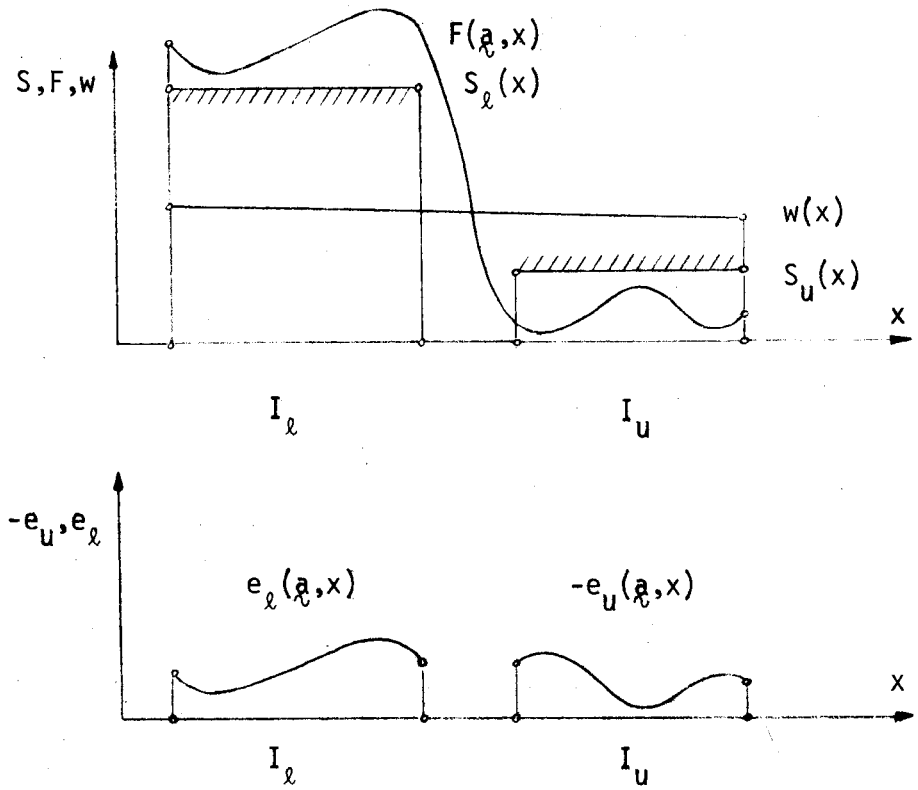


Fig. 6-3 Example of a design problem when the specification is satisfied. Case II is applicable.

Now, in an effort to exceed the specification by the greatest amount, the following objective function was proposed [34] to be minimized

$$(6.9) \qquad U(\underset{\sim}{a},\xi)=-\{\sum_{i\in I_u}[-e_{u_i}'(\underset{\sim}{a},\xi)]^{-p}+\sum_{i\in I_\ell}[e_{\ell_i}'(\underset{\sim}{a},\xi)]^{-p}\}^{-\frac{1}{p}}$$

for

$$-e_{u_i}'(\underset{\sim}{a},\xi)>0 \ , \quad i\in I_u$$

(6.10)

$$e_{\ell_i}'(\underset{\sim}{a},\xi)>0 \ , \quad i\in I_\ell$$

and

$$p\geq 1.$$

Again, the larger the value of p the more nearly would we expect the minimum error to be emphasized, since

$$\max_{\substack{i\in I_u \\ j\in I_\ell}}[e_{u_i}'(\underset{\sim}{a},\xi), -e_{\ell_j}'(\underset{\sim}{a},\xi)]=\lim_{p\to\infty} U(\underset{\sim}{a},\xi), \tag{6.11}$$

6.4 Scaling

If p is very large we have ill-conditioning in both cases resulting from the numerical evaluation of $[\pm e_{u_i}']^{\pm p}$ and $[\mp e_{\ell_i}']^{\pm p}$. By proper scaling we can alleviate the ill-conditioning and also define only <u>one</u> objective function which is valid for both cases [34].

Let

$$M(\underset{\sim}{a},\xi)\overset{\Delta}{=}\max_{\substack{i\in I_u \\ j\in I_\ell}}[e_{u_i}'(\underset{\sim}{a},\xi), -e_{\ell_j}'(\underset{\sim}{a},\xi)], \tag{6.12}$$

and define an objective function

$$U(\underset{\sim}{a},\xi)=M(\underset{\sim}{a},\xi)\Big(\sum_{i\in K_u}\Big[\frac{e_{u_i}'(\underset{\sim}{a},\xi)}{M(\underset{\sim}{a},\xi)}\Big]^q+\sum_{i\in K_\ell}\Big[\frac{-e_{\ell_i}'(\underset{\sim}{a},\xi)}{M(\underset{\sim}{a},\xi)}\Big]^q\Big)^{\frac{1}{q}} \tag{6.13}$$

and its gradients

$$\nabla U(\underset{\sim}{a},\xi)=\Big(\sum_{i\in K_u}\Big[\frac{e_{u_i}'(\underset{\sim}{a},\xi)}{M(\underset{\sim}{a},\xi)}\Big]^q+\sum_{i\in K_\ell}\Big[\frac{-e_{\ell_i}'(\underset{\sim}{a},\xi)}{M(\underset{\sim}{a},\xi)}\Big]^q\Big)^{\frac{1}{q}-1}$$

$$\cdot\Big(\sum_{i\in K_u}\Big[\frac{e_{u_i}'(\underset{\sim}{a},\xi)}{M(\underset{\sim}{a},\xi)}\Big]^{q-1}\nabla e_{u_i}'(\underset{\sim}{a},\xi)$$

$$-\sum_{i\in K_\ell}\Big[\frac{-e_{\ell_i}'(\underset{\sim}{a},\xi)}{M(\underset{\sim}{a},\xi)}\Big]^{q-1}\nabla e_{\ell_i}'(\underset{\sim}{a},\xi)\Big) \tag{6.14}$$

where

$$K_u\triangleq\begin{cases}J_u(\underset{\sim}{a},\xi) & \text{if } M(\underset{\sim}{a},\xi)>0\\ I_u & \text{if } M(\underset{\sim}{a},\xi)<0\end{cases},$$

$$K_\ell\triangleq\begin{cases}J_\ell(\underset{\sim}{a},\xi) & \text{if } M(\underset{\sim}{a},\xi)>0\\ I_\ell & \text{if } M(\underset{\sim}{a},\xi)<0\end{cases}, \tag{6.15}$$

with $J_u(\underset{\sim}{a},\xi)$ and $J_\ell(\underset{\sim}{a},\xi)$ defined in (6.7) and I_u and I_ℓ defined in (6.10), and

$$q\triangleq p\ \operatorname{sgn} M(\underset{\sim}{a},\xi)\begin{cases}p>1 \text{ for } M(\underset{\sim}{a},\xi)>0,\\ p\geq 1 \text{ for } M(\underset{\sim}{a},\xi)<0.\end{cases} \tag{6.16}$$

If $e_{u_i}'(\underset{\sim}{a},\xi)$ and $e_{\ell_i}'(\underset{\sim}{a},\xi)$ for $i\in I_u$ and $i\in I_\ell$, respectively, are continuous with continuous partial derivatives, the proposed objective function is continuous everywhere with continuous partial derivatives. The objective function (6.13) and partial derivatives (6.14) still remain continuous even when, for some i's, e_{u_i}' and/or e_{ℓ_i}' are discontinuous or continuous with discontinuous derivatives, simply because those points are ignored if e_{u_i}'/M and/or $-e_{\ell_i}'/M$ are negative and $M>0$. This is very suitable for the wide variety of network and system design problems, especially in filter design.

6.5 Discussion

The artificial margin ξ which is constant during optimization does not affect the location of the minimax optimum ($p \to \infty$). Its important role, however, is evident for a finite value of p. The value of the parameter ξ can be chosen so that the M of (6.12) is always positive or negative during optimization. When M is positive, only sample points which satisfy the conditions in (6.7) are considered and, therefore, there is a saving in gradient computation. But in this case it may happen that M=0, when the function is continuous but the derivatives are discontinuous. On the rare occasions when this situation causes a failure of the gradient minimization algorithm, one can change the value of ξ and restart the optimization process. If the value of M is chosen to be negative this possible failure is avoided.

6.6 Computer program

A computer program in FORTRAN IV has been written, which utilizes the least pth approach described in previous sections. The program consists of 15 separate subprograms.

FMLPO Supplies all the data for the optimization process and coordinates the other subprograms in the package;

FUNCS Defines upper and lower specified functions;

FCTAPP Defines an approximating function and its gradients

with respect to variable parameters;

W Defines upper and lower weighting functions;

FCT Calculates artificial upper and lower specified functions according to the equation (6.3);

EPSNP Calculates upper and lower weighted error functions;

ERRO Selects the weighted error functions of interest for the objective function (6.13) according to the equations given in (6.15);

FUNCG Computes the generalized least pth objective function (6.13) and its gradients (6.14) w.r.t. variable parameters;

GRDGHK, FMFPG and FMNFG are subroutine subprograms which have the same role as the subroutines GRDCHK, FMFPC and FMNFC, respectively, in the FMCLP package.

INPUT, FINAL, WRITE1 and WRITE2 are subroutine subprograms which have already been introduced in section 5.3.

FUNCS, W, FCT and EPSNP are function subprograms, and the others are subroutine subprograms.

A user of the package is supposed to write the following subprograms by himself: FUNCS, FCTAPP and W in a straightforward way. More detailed instructions may be found in Appendix V. A user has to arrange the intervals, not necessarily disjoint, such

that each of them has only one specification. For example, if the original design problem has upper and lower specifications for the same values of the independent parameter x, two intervals with a single specification have to be formed, one with the upper and the other with the lower specification. A two-dimensional array is constructed of the input data, where the first of the two subscripts relates to the type of the specification and the second denotes the appropriate intervals. The elements of the array may have values +1 if there is an upper or -1 if there is a lower specification in a given interval.

Let

$$(6.17)\qquad \begin{aligned} e_i''(\underset{\sim}{a},\xi) &\triangleq (+1)\cdot e_{u_i}'(\underset{\sim}{a},\xi) \quad , \quad i\in I_u \\ e_i''(\underset{\sim}{a},\xi) &\triangleq (-1)\cdot e_{\ell_i}'(\underset{\sim}{a},\xi) \quad , \quad i\in I_\ell \end{aligned}$$

where $I_u \cap I_\ell = \emptyset$.

Then the expressions for the generalized least pth approximation (6.13)-(6.16) may be redefined in a form suitable for programming:

$$(6.18)\qquad M(\underset{\sim}{a},\xi)\triangleq \max_i [e_i''(\underset{\sim}{a},\xi)] \quad , \quad i\in I \triangleq I_u \cup I_\ell$$

$$(6.19)\qquad U(\underset{\sim}{a},\xi)=M(\underset{\sim}{a},\xi)\Big(\sum_{i\in K}\Big[\frac{e_i''(\underset{\sim}{a},\xi)}{M(\underset{\sim}{a},\xi)}\Big]^q\Big)^{\frac{1}{q}}$$

$$(6.20)\qquad \underset{\sim}{\nabla}U(\underset{\sim}{a},\xi)=\Big(\sum_{i\in K}\Big[\frac{e_i''(\underset{\sim}{a},\xi)}{M(\underset{\sim}{a},\xi)}\Big]^q\Big)^{\frac{1}{q}-1}\cdot\Big(\sum_{i\in K}\Big[\frac{e_i''(\underset{\sim}{a},\xi)}{M(\underset{\sim}{a},\xi)}\Big]^{q-1}\underset{\sim}{\nabla}e_i''(\underset{\sim}{a},\xi)\Big)$$

where

$$(6.21)\qquad K\triangleq\begin{cases} J(\underset{\sim}{a},\xi) & \text{if } M(\underset{\sim}{a},\xi)>0 \\ I & \text{if } M(\underset{\sim}{a},\xi)<0 \end{cases}$$

with $J \overset{\Delta}{=} J_u(\underset{\sim}{a},\xi) \cup J_\ell(\underset{\sim}{a},\xi)$, and q the same as in (6.16).

Function subprogram EPSNP computes the upper or lower weighted error at a single point x for a current variable vector $\underset{\sim}{a}$. A flowchart of function EPSNP is shown in Fig. 6-5.

The flowcharts of the subroutines ERRO and FUNCG are available in Fig. 6-6 and Fig. 6-7, respectively. The list of the symbols used in the flowcharts are given on the page 67. A scheme presented in Fig. 6-4 illustrates how the subprograms are connected to each other. The flowchart of FMLPO is given in Fig. 6-8.

If the gradients of the approximating function are not supplied correctly, the program will terminate and print out the appropriate message. Also, suitable diagnostic messages are printed out whenever there is any unusual exit.

When the criteria for the optimum have been satisfied, the optimum solution, the discrete point set with the articifial weighted errors in each interval and the execution time in seconds are printed out.

The computer program of the whole package FMLPO is given in Appendices III and VI.

Most of the symbols used in the flowchart are in agreement with the corresponding ones used in the theory. Those which differ or have not been mentioned yet are

$error_i$ corresponds to $e_i(\underset{\sim}{a},\varepsilon)$;

e_{max} corresponds to $M(\underset{\sim}{a},\varepsilon)$;

n_I is a total number of intervals;

j_I is a current interval;

k_j is a number of discrete point set at (j_I)th interval;

$x_{1,j}$ is a left end point of (j_I)th interval

$x_{2,j}$ is a right end point of (j_I)th interval;

$x_{3,j}$ is a characteristic number of (j_I)th interval with the information about upper or lower specification;

m_j is an integer which determinates the interval where the selected point belongs;

$grad_i$ is the gradient of the approximating function $\underset{\sim}{\nabla}F(\underset{\sim}{a},x_i)$;

ε is a small quantity used as a stopping criterion of the iterative procedure in FMLPO.

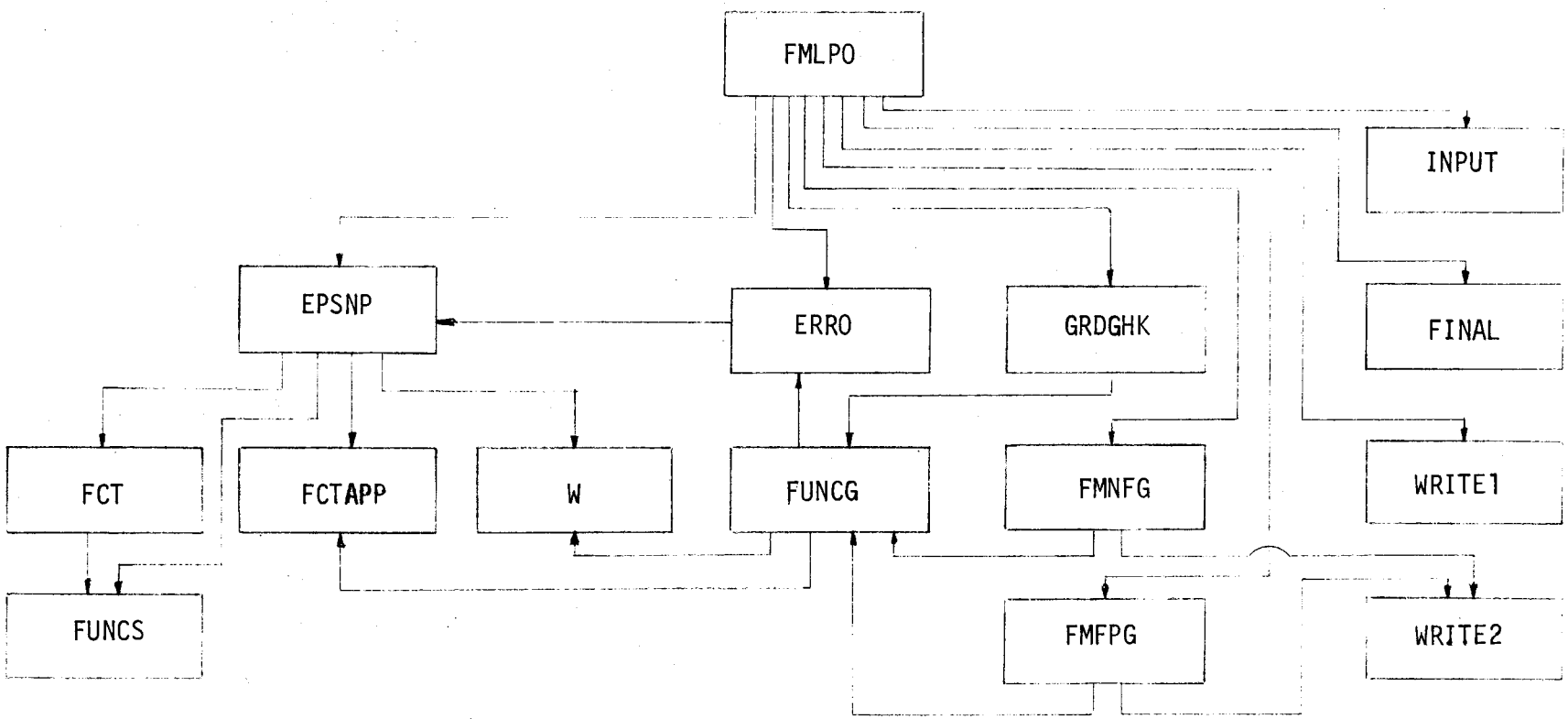


Fig. 6-4 The organization of FMLPO

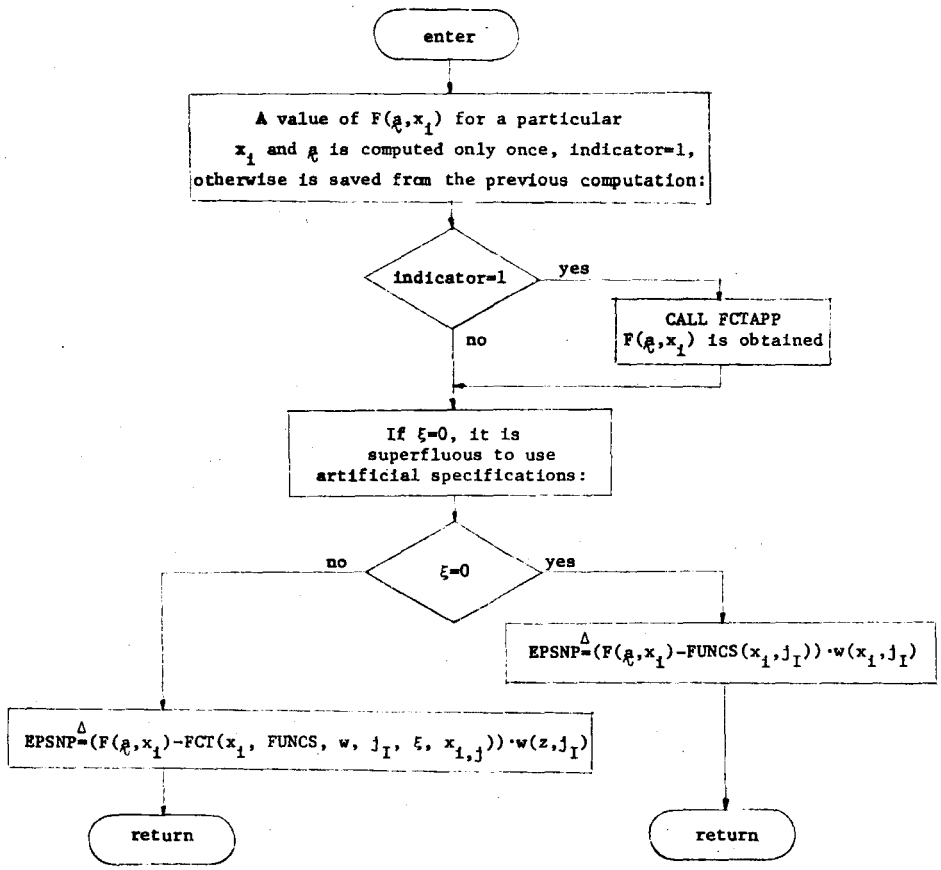
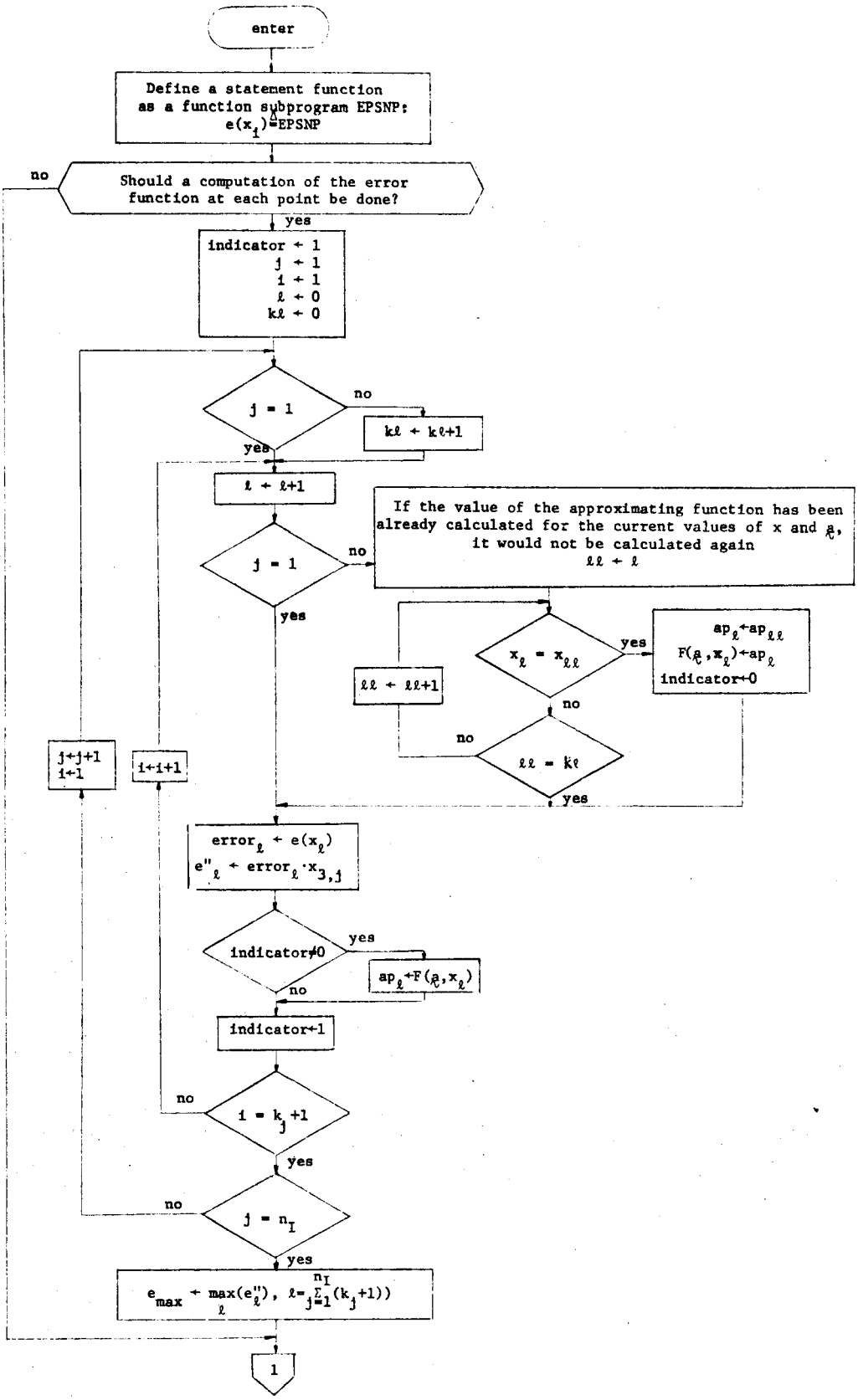


Fig. 6-5 Flowchart of function subprogram EPSNP


enter
Define a statement function as a function subprogram EPSNP: e(x_l)=EPSNP
Should a computation of the error function at each point be done?
no
yes
indicator ← 1
j ← 1
i ← 1
ℓ ← 0
kℓ ← 0
j = 1
no
yes
kℓ ← kℓ+1
ℓ ← ℓ+1
j = 1
no
yes
If the value of the approximating function has been already calculated for the current values of x and a, it would not be calculated again
ℓℓ ← ℓ
x_ℓ = x_ℓℓ
yes
no
ap_ℓ ← ap_ℓℓ
F(a,x_ℓ) ← ap_ℓ
indicator ← 0
ℓℓ ← ℓℓ+1
ℓℓ = kℓ
no
yes
j ← j+1
i ← 1
i ← i+1
error_ℓ ← e(x_ℓ)
e''_ℓ ← error_ℓ · x_3,j
indicator≠0
yes
no
ap_ℓ ← F(a,x_ℓ)
indicator ← 1
i = k_j +1
no
yes
j = n_I
no
yes
e_max ← max_ℓ(e''_ℓ), ℓ = Σ_{j=1}^{n_I}(k_j+1))
1

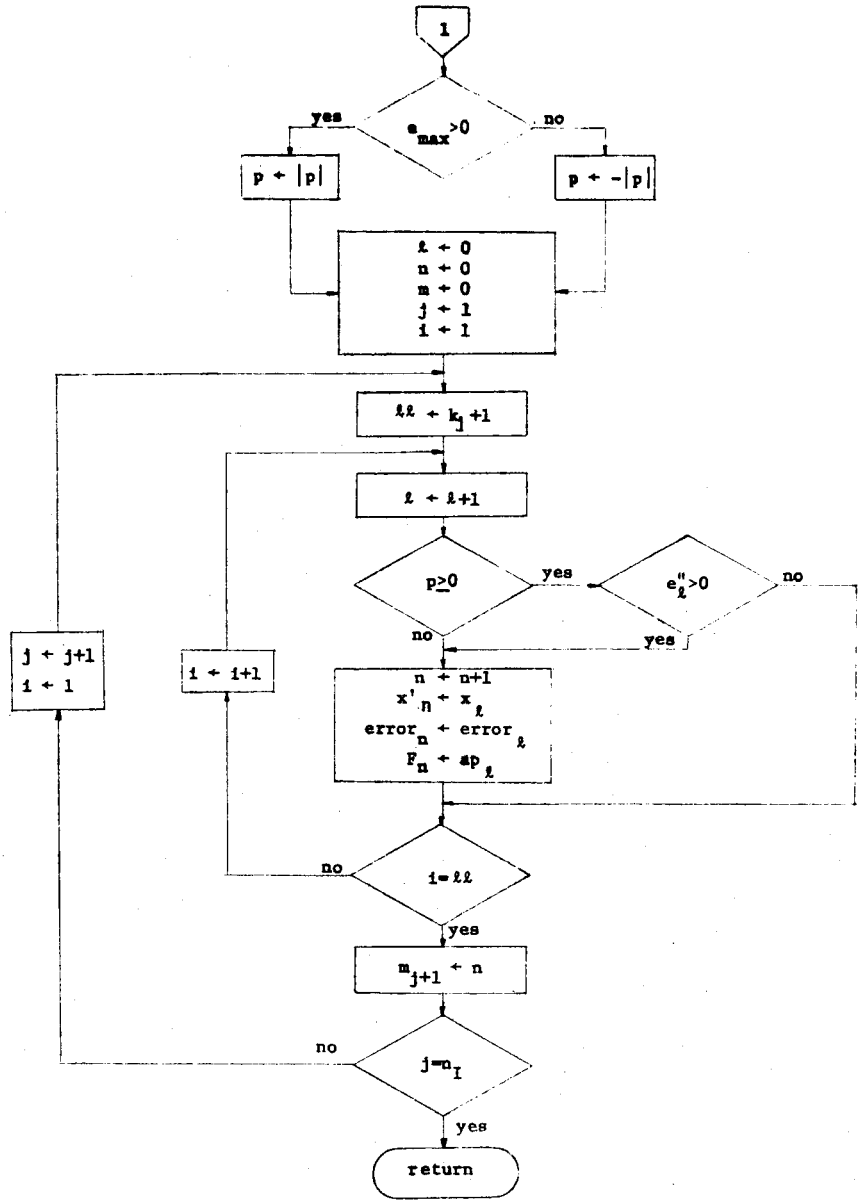
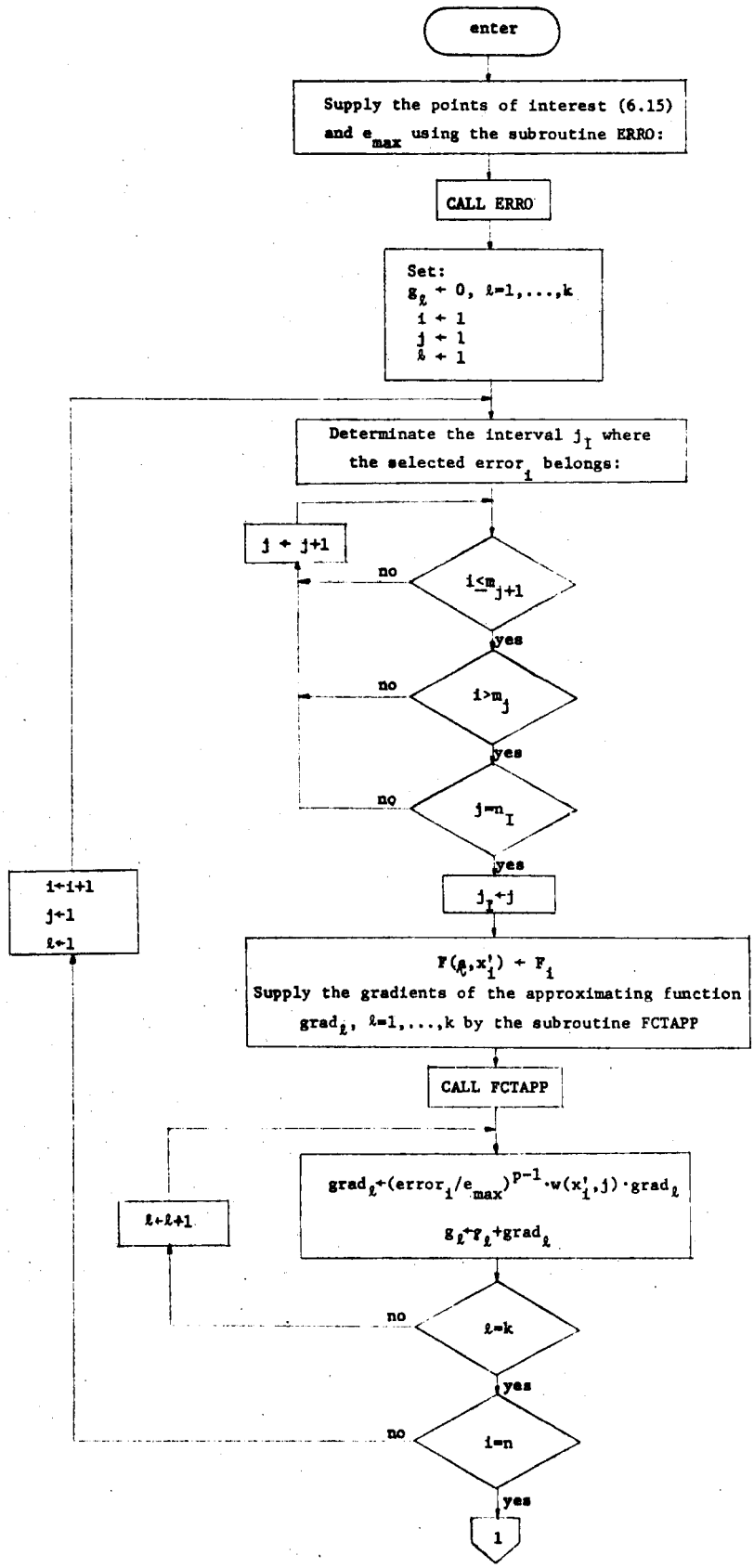


Fig. 6-6 Flowchart of subroutine ERRO


enter
Supply the points of interest (6.15) and e_{max} using the subroutine ERRO:
CALL ERRO
Set:
$g_\ell \leftarrow 0$, $\ell=1,\ldots,k$
$i \leftarrow 1$
$j \leftarrow 1$
$\ell \leftarrow 1$
Determinate the interval j_I where the selected $error_i$ belongs:
$j \leftarrow j+1$
no
$i \leq m_{j+1}$
yes
no
$i > m_j$
yes
no
$j=n_I$
yes
$j_I \leftarrow j$
$i \leftarrow i+1$
$j \leftarrow 1$
$\ell \leftarrow 1$
$F(\ell, x_i') \leftarrow F_i$
Supply the gradients of the approximating function
$grad_\ell$, $\ell=1,\ldots,k$ by the subroutine FCTAPP
CALL FCTAPP
$grad_\ell \leftarrow (error_i/e_{max})^{p-1} \cdot w(x_i', j) \cdot grad_\ell$
$g_\ell \leftarrow g_\ell + grad_\ell$
$\ell \leftarrow \ell+1$
no
$\ell=k$
yes
no
$i=n$
yes
1

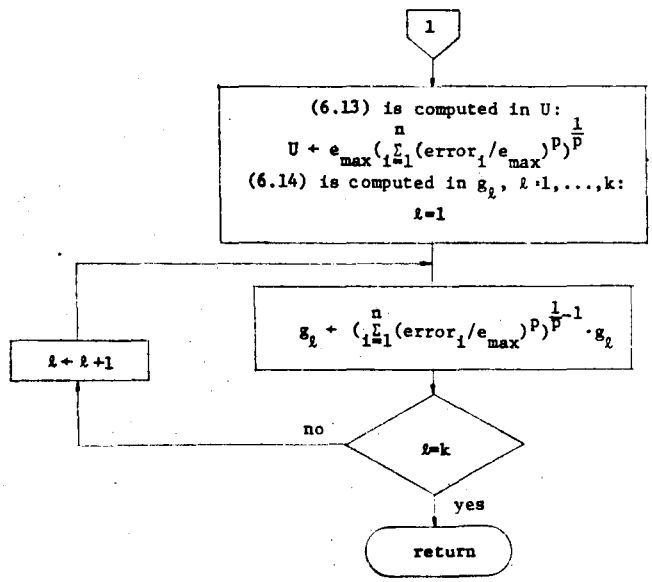
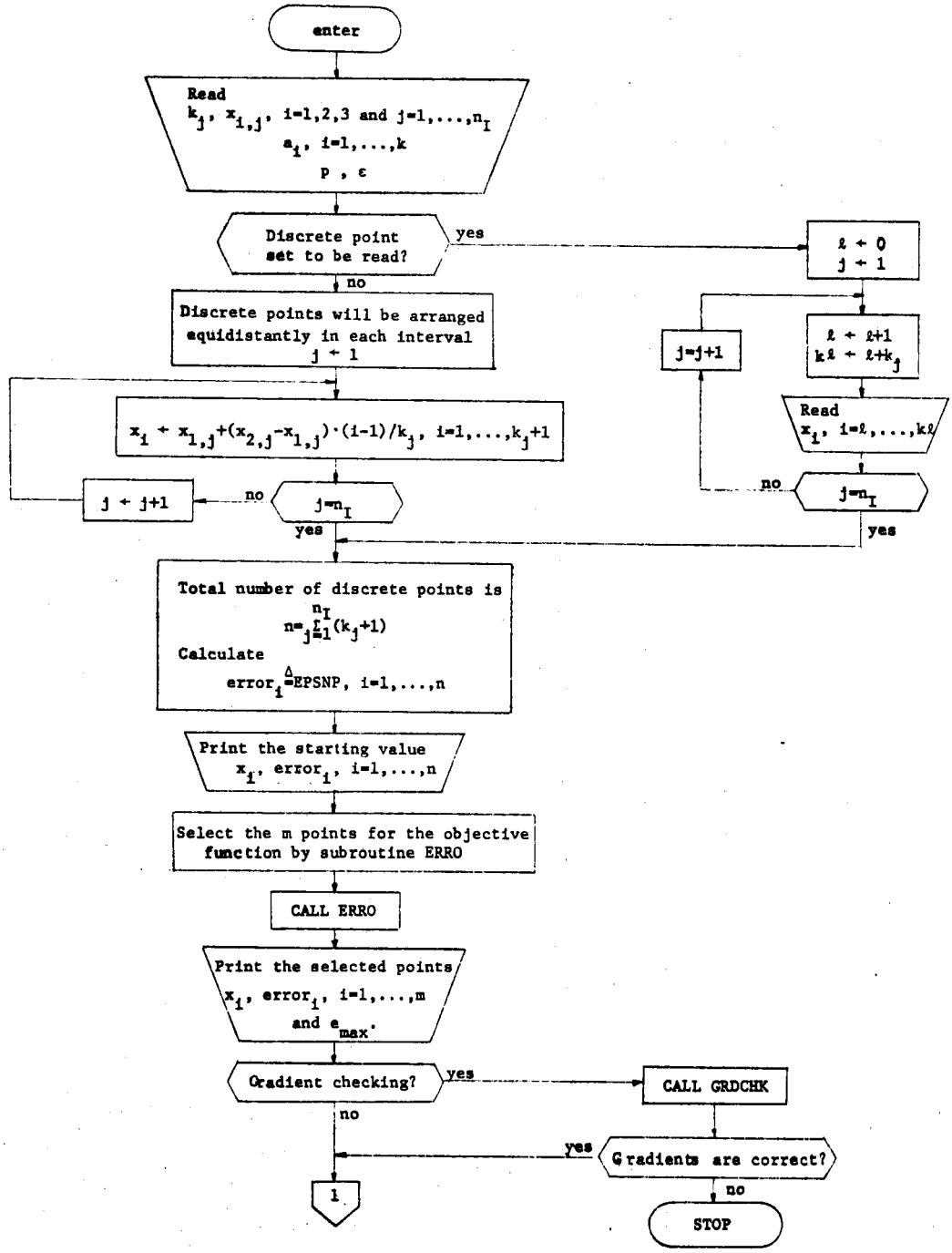


Fig. 6-7 Flowchart of subroutine FUNCG


enter
Read
k_j, $x_{i,j}$, i=1,2,3 and j=1,...,n_I
a_i, i=1,...,k
p , ε
Discrete point set to be read?
yes
no
ℓ ← 0
j ← 1
Discrete points will be arranged equidistantly in each interval
j ← 1
j=j+1
ℓ ← ℓ+1
kℓ ← ℓ+k_j
Read
x_i, i=ℓ,...,kℓ
$x_i \leftarrow x_{1,j}+(x_{2,j}-x_{1,j})\cdot(i-1)/k_j$, i=1,...,$k_j$+1
j ← j+1
no
j=n_I
yes
no
j=n_I
yes
Total number of discrete points is
$n=\sum_{j=1}^{n_I}(k_j+1)$
Calculate
$error_i \overset{\Delta}{=} EPSNP$, i=1,...,n
Print the starting value
x_i, $error_i$, i=1,...,n
Select the m points for the objective function by subroutine ERRO
CALL ERRO
Print the selected points
x_i, $error_i$, i=1,...,m
and e_{max}.
Gradient checking?
yes
CALL GRDCHK
no
yes
Gradients are correct?
no
STOP
1

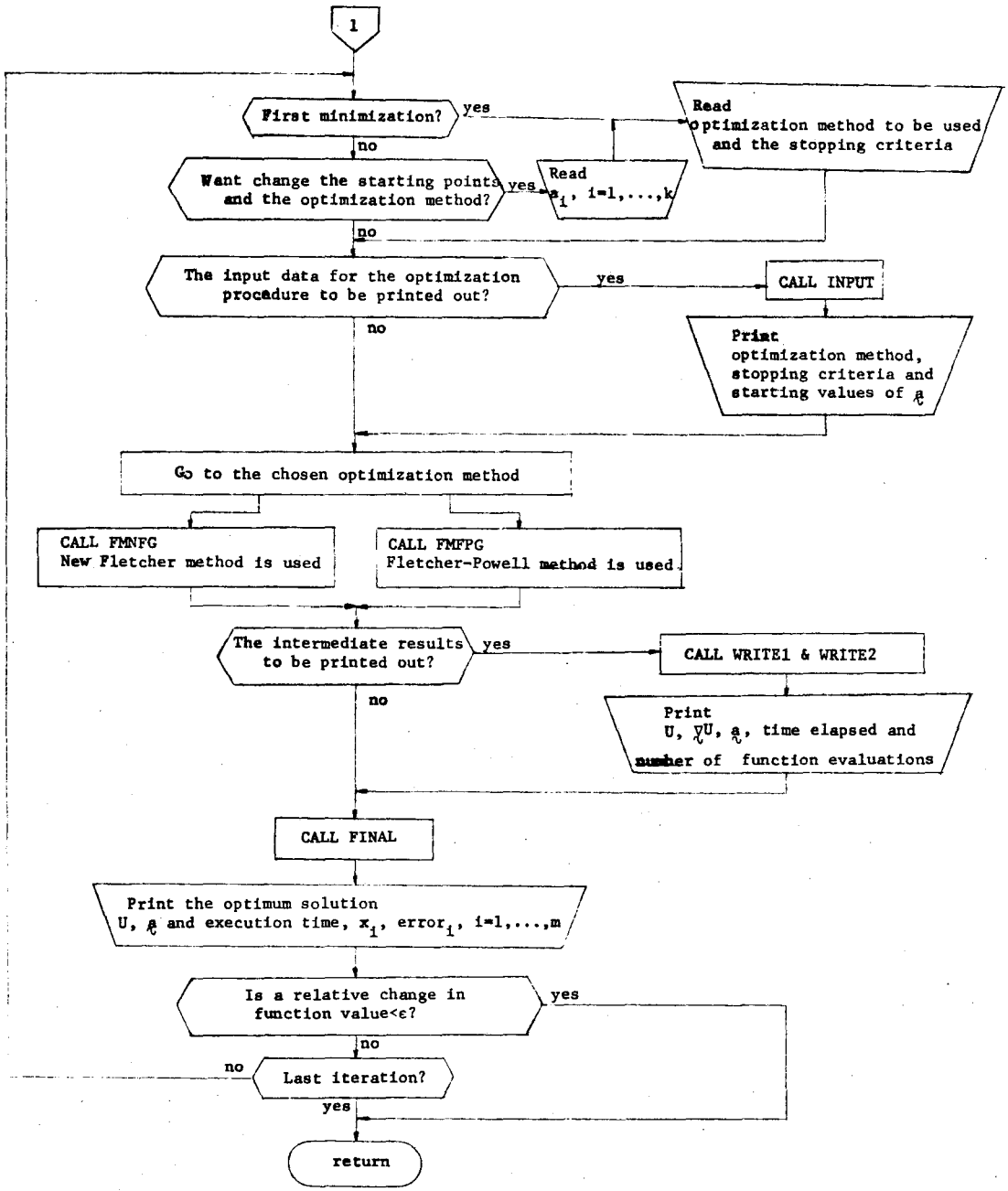


Fig. 6-8 Flowchart of subroutine FMLPO

6.7 Examples

Four examples were chosen to illustrate the work of the package. The first example is with the objective function as in case I, the second when it is as in case II, the third the low pass filter design problem without and, fourth, with parameter constraints.

Test 1

A problem for system modelling, the same as used in Test 4 in Chapter V, is solved again. In this case upper and lower specified functions are chosen to be the same:

$$(6.22) \qquad S_u'(t)=S_\ell'(t)=S(t)=\frac{3}{20}e^{-t}+\frac{1}{52}e^{-5t}-\frac{1}{65}e^{-2t}(3\sin 2t+11\cos 2t)$$

with the artificial margin $\xi=0$, over the range $t\varepsilon[0,10]$.

When one function approximates the other, it always corresponds to the situation in the case I.

We consider the same interval I twice, once as I_u and secondly as I_ℓ. 51 uniformly spaced points were chosen over the range [0,10] including 0 and 10, and a starting vector

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} .$$

The Fletcher-Powell and Fletcher optimization techniques were used with the stopping criterion $\varepsilon=10^{-6}$. The results are

shown in Table 6-2. Also, the list of errors in both intervals and selected errors for further optimization, for p=2 is given in Table 6-1. With p=2, the following parameter vector is found

$$\begin{bmatrix} 1.01647 \\ 0.78927 \\ 0.161400 \end{bmatrix} .$$

The results from Table 6-2 are in excellent agreement with those in Table 5-6 when n=50 and n_s=1, i.e. the case without quadratic interpolation. Also it is evident that the Fletcher method is more efficient.

Test 2

The same optimization problem was chosen as in the Test 1, but this time the artificial margin is set to be different than zero, in particular

$$\xi = 2 \times 10^{-2}$$

therefore the artificial upper and lower specifications are distinct. Two possible cases may occur now: the specification may be violated and the specification may be satisfied.

The same initial set of sampling points and the starting value $\underset{\sim}{a}$ as in Test 1 was used. From the Table 6-3, we may conclude that the starting objective function belongs to case I. The Fletcher optimization method is used with the same stopping criterion as in the Test 1 and already for p=2 the problem is in case II, also is illustrated

sample points	independent variable in I	error: upper=lower	selected errors: nonnegative upper & nonpositive lower
1	0.0	$-8.882\ 10^{-16}$	$1.288\ 10^{-2}$
2	0.2	$1.288\ 10^{-2}$	$6.985\ 10^{-3}$
3	0.4	$6.985\ 10^{-3}$	$1.033\ 10^{-3}$
4	0.6	$-1.173\ 10^{-3}$	$2.944\ 10^{-3}$
5	0.8	$-5.906\ 10^{-3}$	$3.810\ 10^{-3}$
6	1.0	$-6.635\ 10^{-3}$	$3.755\ 10^{-3}$
7	1.2	$-4.701\ 10^{-3}$	$3.043\ 10^{-3}$
8	1.4	$-1.736\ 10^{-3}$	$1.970\ 10^{-4}$
9	1.6	$1.033\ 10^{-3}$	$7.867\ 10^{-3}$
10	1.8	$2.944\ 10^{-3}$	$7.341\ 10^{-7}$
11	2.0	$3.810\ 10^{-3}$	$1.215\ 10^{-6}$
12	2.2	$3.755\ 10^{-3}$	$1.263\ 10^{-6}$
13	2.4	$3.043\ 10^{-3}$	$1.058\ 10^{-6}$
14	2.6	$1.970\ 10^{-3}$	$-8.882\ 10^{-16}$
15	2.8	$7.868\ 10^{-4}$	$-1.173\ 10^{-3}$
16	3.0	$-3.253\ 10^{-4}$	$-5.906\ 10^{-3}$
17	3.2	$-1.258\ 10^{-3}$	$-6.635\ 10^{-3}$
18	3.4	$-1.963\ 10^{-3}$	$-4.701\ 10^{-3}$
19	3.6	$-2.437\ 10^{-3}$	$-1.736\ 10^{-3}$
20	3.8	$-2.702\ 10^{-3}$	$-3.253\ 10^{-4}$
21	4.0	$-2.795\ 10^{-3}$	$-1.258\ 10^{-3}$
22	4.2	$-2.754\ 10^{-3}$	$-1.963\ 10^{-3}$
23	4.4	$-2.617\ 10^{-3}$	$-2.437\ 10^{-3}$
24	4.6	$-2.419\ 10^{-3}$	$-2.702\ 10^{-3}$
25	4.8	$-2.183\ 10^{-3}$	$-2.795\ 10^{-3}$

sample points	independent variable in I	error: upper=lower	selected errors: nonnegative upper & nonpositive lower
26	5.0	$-1.933\ 10^{-3}$	$-2.754\ 10^{-3}$
27	5.2	$-1.681\ 10^{-3}$	-2.617×10^{-3}
28	5.4	$-1.440\ 10^{-3}$	$-2.419\ 10^{-3}$
29	5.6	$-1.215\ 10^{-3}$	$-2.183\ 10^{-3}$
30	5.8	$-1.011\ 10^{-3}$	$-1.933\ 10^{-3}$
31	6.0	$-8.301\ 10^{-4}$	$-1.681\ 10^{-3}$
32	6.2	$-6.723\ 10^{-4}$	$-1.440\ 10^{-3}$
33	6.4	$-5.371\ 10^{-4}$	$-1.215\ 10^{-3}$
34	6.6	$-4.231\ 10^{-4}$	$-1.011\ 10^{-3}$
35	6.8	$-3.284\ 10^{-4}$	$-8.301\ 10^{-4}$
36	7.0	$-2.510\ 10^{-4}$	$-6.723\ 10^{-4}$
37	7.2	$-1.886\ 10^{-4}$	$-5.371\ 10^{-4}$
38	7.4	$-1.391\ 10^{-4}$	$-4.231\ 10^{-4}$
39	7.6	$-1.005\ 10^{-4}$	$-3.284\ 10^{-4}$
40	7.8	$-7.081\ 10^{-5}$	$-2.510\ 10^{-4}$
41	8.0	$-4.848\ 10^{-5}$	$-1.886\ 10^{-4}$
42	8.2	$-3.199\ 10^{-5}$	$-1.391\ 10^{-4}$
43	8.4	$-2.013\ 10^{-5}$	$-1.005\ 10^{-4}$
44	8.6	$-1.182\ 10^{-5}$	$-7.081\ 10^{-5}$
45	8.8	$-6.211\ 10^{-6}$	$-4.848\ 10^{-5}$
46	9.0	$-2.597\ 10^{-6}$	$-3.199\ 10^{-5}$
47	9.2	$-4.265\ 10^{-7}$	$-2.013\ 10^{-5}$
48	9.4	$7.341\ 10^{-7}$	$-1.182\ 10^{-5}$
49	9.6	$1.215\ 10^{-6}$	$-6.211\ 10^{-6}$
50	9.8	$1.263\ 10^{-6}$	$-2.597\ 10^{-6}$
51	10.0	$1.058\ 10^{-6}$	$-4.265\ 10^{-7}$

Table 6-1

	p=2		p=10^2	
	Fletcher Powell	Fletcher	Fletcher-Powell	Fletcher
a_1	1.01647063	1.01647060	6.8785026×10^{-1}	6.8785037×10^{-1}
a_2	7.8927025×10^{-1}	7.8927036×10^{-1}	9.5318185×10^{-1}	9.5318177×10^{-1}
a_3	1.61400086×10^{-1}	1.61400093×10^{-1}	1.2318597×10^{-1}	1.2318599×10^{-1}
$M(\underset{\sim}{a})$	1.2880047×10^{-2}	1.2880048×10^{-2}	7.988425×10^{-3}	7.988427×10^{-3}
t_M	2.0×10^{-1}	2.0×10^{-1}	2.0×10^{-1}	2.0×10^{-1}
$U(\underset{\sim}{a})$	2.09004705×10^{-2}	2.09004705×10^{-2}	8.04667205×10^{-3}	8.04667205×10^{-3}
execution time in seconds	7.5	7.2	10	7.4
	p=10		p=10^4	
	Fletcher-Powell	Fletcher	Fletcher-Powell	Fletcher
a_1	7.4276852×10^{-1}	7.4276856×10^{-1}	6.844480×10^{-1}	6.844475×10^{-1}
a_2	9.295083×10^{-1}	9.295085×10^{-1}	9.540875×10^{-1}	9.540873×10^{-1}
a_3	1.2803127×10^{-1}	1.2803137×10^{-1}	1.2286721×10^{-1}	1.2286716×10^{-1}
$M(\underset{\sim}{a})$	8.592078×10^{-3}	8.59209×10^{-3}	7.947445×10^{-3}	7.947439×10^{-3}
t_M	2.0×10^{-1}	2.0×10^{-1}	2.0×10^{-1}	2.0×10^{-1}
$U(\underset{\sim}{a})$	9.22275978×10^{-3}	9.22275978×10^{-3}	7.94802468×10^{-3}	7.94802476×10^{-3}
execution time in seconds	13.7	6.7	8.6	6.8

Table 6-2

in Table 6-3. The optimal solutions for different values of p are presented in Table 6-4. The artificial margin ξ does not affect the optimal solution of the variable vector $\underset{\sim}{a}$. It has the same value as the corresponding vector from the Table 6-2.

The starting vector $\underset{\sim}{a}$ was

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

and the following vector was found

$$\begin{bmatrix} 0.92369 \\ 0.83516 \\ 0.14720 \end{bmatrix}$$

for p=2.

Test 3

The computer program was used in the optimization of a five section cascaded transmission-line low pass filter which has been considered by Carlin [35]. The terminations of the filter are unity, the length of the ith section ℓ_i and the normalized characteristic impedance of the ith section Z_{oi}, such that a maximum insertion loss γ in the passband, from 0 to 1 GHz, is not more than 0.4 dB, while maximizing γ at a point in the stopband. All section lengths were kept fixed at 2.5 cm so that

$\xi=2\times10^{-2}$	Initial values		p=2
sample points	upper errors; lower=upper+2ξ	selected errors nonnegative upper nonpositive lower	upper errors; lower=upper+2ξ
1	-2.000×10^{-2}		-2.000×10^{-2}
2	1.293×10^{-1}	1.293×10^{-1}	-8.991×10^{-3}
3	2.057×10^{-1}	2.057×10^{-1}	-1.535×10^{-2}
4	2.380×10^{-1}	2.380×10^{-1}	-2.321×10^{-2}
5	2.429×10^{-1}	2.429×10^{-1}	-2.728×10^{-2}
6	2.304×10^{-1}	2.304×10^{-1}	-2.727×10^{-2}
7	2.070×10^{-1}	2.070×10^{-1}	-2.469×10^{-2}
8	1.772×10^{-1}	1.772×10^{-1}	-2.124×10^{-2}
9	1.445×10^{-1}	1.445×10^{-1}	-1.818×10^{-2}
10	1.115×10^{-1}	1.115×10^{-1}	-1.615×10^{-2}
11	8.009×10^{-2}	8.009×10^{-2}	-1.532×10^{-2}
12	5.178×10^{-2}	5.178×10^{-2}	-1.551×10^{-2}
13	2.741×10^{-2}	2.741×10^{-2}	-1.643×10^{-2}
14	7.360×10^{-3}	7.360×10^{-3}	-1.776×10^{-2}
15	-8.373×10^{-3}		-1.919×10^{-2}
16	-2.007×10^{-2}		-2.054×10^{-2}
17	-2.820×10^{-2}		-2.168×10^{-2}
18	-3.334×10^{-2}		-2.255×10^{-2}
19	-3.608×10^{-2}		-2.314×10^{-2}
20	-3.699×10^{-2}		-2.348×10^{-2}
21	-3.660×10^{-2}		-2.360×10^{-2}
22	-3.533×10^{-2}		-2.356×10^{-2}
23	-3.354×10^{-2}		-2.340×10^{-2}
24	-3.151×10^{-2}		-2.314×10^{-2}
25	-2.944×10^{-2}		-2.284×10^{-2}

CONT'D

sample points	upper errors; lower=upper+2ξ	selected errors nonnegative upper nonpositive lower	upper errors; lower=upper+2ξ
26	-2.748×10^{-2}		-2.251×10^{-2}
27	-2.570×10^{-2}		-2.218×10^{-2}
28	-2.417×10^{-2}		-2.186×10^{-2}
29	-2.289×10^{-2}		-2.155×10^{-2}
30	-2.186×10^{-2}		-2.128×10^{-2}
31	-2.106×10^{-2}		-2.103×10^{-2}
32	-2.047×10^{-2}		-2.082×10^{-2}
33	-2.005×10^{-2}		-2.063×10^{-2}
34	-1.978×10^{-2}		-2.048×10^{-2}
35	-1.962×10^{-2}		-2.035×10^{-2}
36	-1.954×10^{-2}		-2.025×10^{-2}
37	-1.952×10^{-2}		-2.017×10^{-2}
38	-1.954×10^{-2}		-2.011×10^{-2}
39	-1.960×10^{-2}		-2.006×10^{-2}
40	-1.965×10^{-2}		-2.003×10^{-2}
41	-1.972×10^{-2}		-2.000×10^{-2}
42	-1.978×10^{-2}		-1.999×10^{-2}
43	-1.984×10^{-2}		-1.998×10^{-2}
44	-1.989×10^{-2}		-1.9978×10^{-2}
45	-1.993×10^{-2}		-1.9977×10^{-2}
46	-1.997×10^{-2}		-1.9977×10^{-2}
47	-1.999×10^{-2}		-1.9979×10^{-2}
48	-2.001×10^{-2}		-1.9982×10^{-2}
49	-2.002×10^{-2}		-1.9986×10^{-2}
50	-2.003×10^{-2}		-1.9989×10^{-2}
51	-2.003×10^{-2}		-1.999×10^{-2}

Table 6-3

Fletcher

	q=-p=-2	q=-p=-10	q=-p=-10^2	q=-p=-10^4
a_1	9.2369×10^{-1}	7.7204×10^{-1}	6.9015×10^{-1}	6.8446×10^{-1}
a_2	8.3515×10^{-1}	9.1379×10^{-1}	9.5224×10^{-1}	9.5408×10^{-1}
a_3	1.4720×10^{-1}	1.3053×10^{-1}	1.2338×10^{-1}	1.2287×10^{-1}
$M(\underset{\sim}{a})$	-8.9912×10^{-3}	-1.1105×10^{-2}	-1.1987×10^{-2}	-1.205×10^{-2}
t_M	0.2	0.2	0.2	0.2
$U(\underset{\sim}{a})$	-1.8963×10^{-3}	-9.9831×10^{-3}	-1.1899×10^{-2}	-1.2051×10^{-2}
function evaluations	33	21	23	33
execution time in seconds	8.4	5.9	6.5	9.8

Table 6-4

the maximum stopband insertion loss would occur at 3GHz and the normalized characteristic impedances are used as variables.

The insertion loss is given by

$$\gamma = 10\log_{10} \frac{1}{1-|\rho|^2} \tag{6.23}$$

where ρ is the reflection coefficient, therefore the corresponding ρ to the insertion loss may be considered in the optimization problem because there is an finite upper bound on $|\rho|$ which is 1. 21 uniformly spaced sample points were used in the passband and a single point at 3GHz. The artificial margin ξ for one case is set to be zero and for the other 0.02337. The weighting function is set to be 1 everywhere. The starting value of the variable vector $\underset{\sim}{a}$ was

$$\begin{bmatrix} 3.180 \\ 0.443 \\ 4.38 \\ 0.443 \\ 3.180 \end{bmatrix}$$

Results obtained using the Fletcher method for the value of $p=10^3$ are presented in Table 6-5 and the response is shown in Fig. 6-9. More details about the computation of the reflection coefficient are given in the Appendix VII.

Although, for physical reasons the symmetrical results for the variables are expected, symmetry was not assumed. ξ does not affect the optimal solution. The reason that it was considered was

Fletcher method

$p=10^3$	$\xi=0$	$\xi=2.337\times10^{-2}$
a_1	3.1525	3.1508
a_2	4.4203×10^{-1}	4.4165×10^{-1}
a_3	4.4212	4.4194
a_4	4.4159×10^{-1}	4.4169×10^{-1}
a_5	3.1526	3.1508
M	3.9466×10^{-5}	-2.3330×10^{-2}
x_M	3.0700×10^{-1}	3.0
U	3.9466×10^{-5}	-2.3330×10^{-2}
Function evaluations	177	79
Execution time in seconds	17	13

Table 6-5

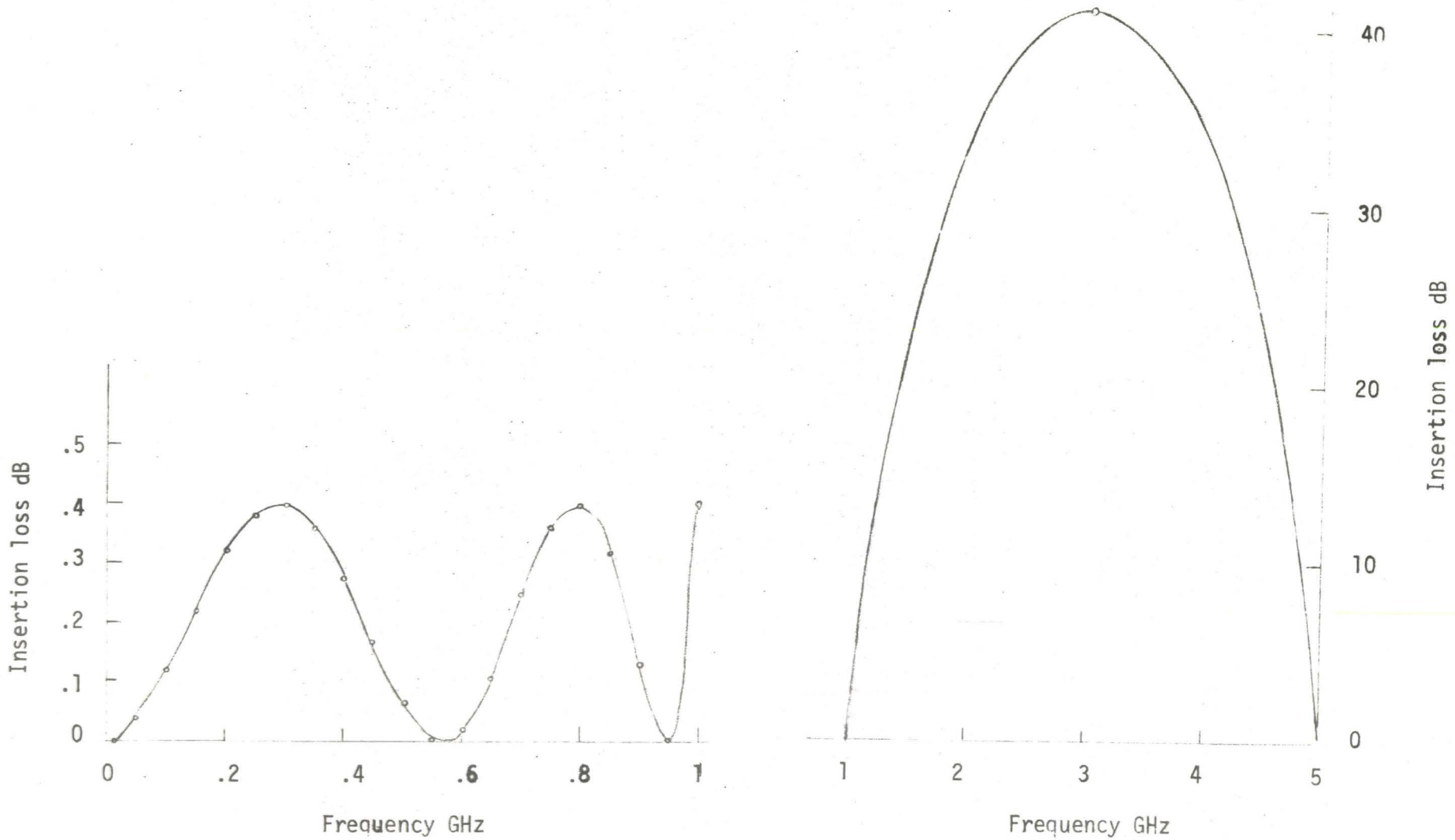


Fig. 6-9 Response of the five section transmission-line filter for the unconstrained design problem.

to bring the objective function into the case when the artificial specification is satisfied.

Test 4

Constraints are put on the parameter vector $\underset{\sim}{a}$ such that they are not satisfied at the optimal solution of the unconstrained problem given in Test 3.

Although the package FMLPO is not written for nonlinear programming, the constrained problem also may be considered. The constraints on a parameter may be considered as upper and lower specifications on an approximating function defined as a single variable parameter over the dummy point outside of the working set of points. This dummy point has to be defined as a new interval for each specification, which is not so practical, but it works. The variable parameter vector $\underset{\sim}{a}$ is constrained by

$$0.2 \leq a_i \leq 4.0, \quad i=1,2,\ldots,5 , \tag{6.24}$$

i.e., the solutions from Table 6-5 for a_3 are infeasible. For the problem given in Test 3 and considering (6.24), results are presented in Table 6-6. The response is shown in Fig. 6-10.

Fletcher method

$p=10^3$	$\xi=0$	$\xi=2.337\times10^{-2}$
a_1	2.9429	2.9422
a_2	4.1069×10^{-1}	4.1070×10^{-1}
a_3	4.0	3.9998
a_4	4.1069×10^{-1}	4.1070×10^{-1}
a_5	2.9429	2.9422
M	4.7403×10^{-5}	-2.3322×10^{-2}
x_M	8.02×10^{-1}	3.0
U	4.7403×10^{-5}	-2.3322×10^{-2}
Function evaluations	55	157
Execution time in seconds	31	55

Table 6-6

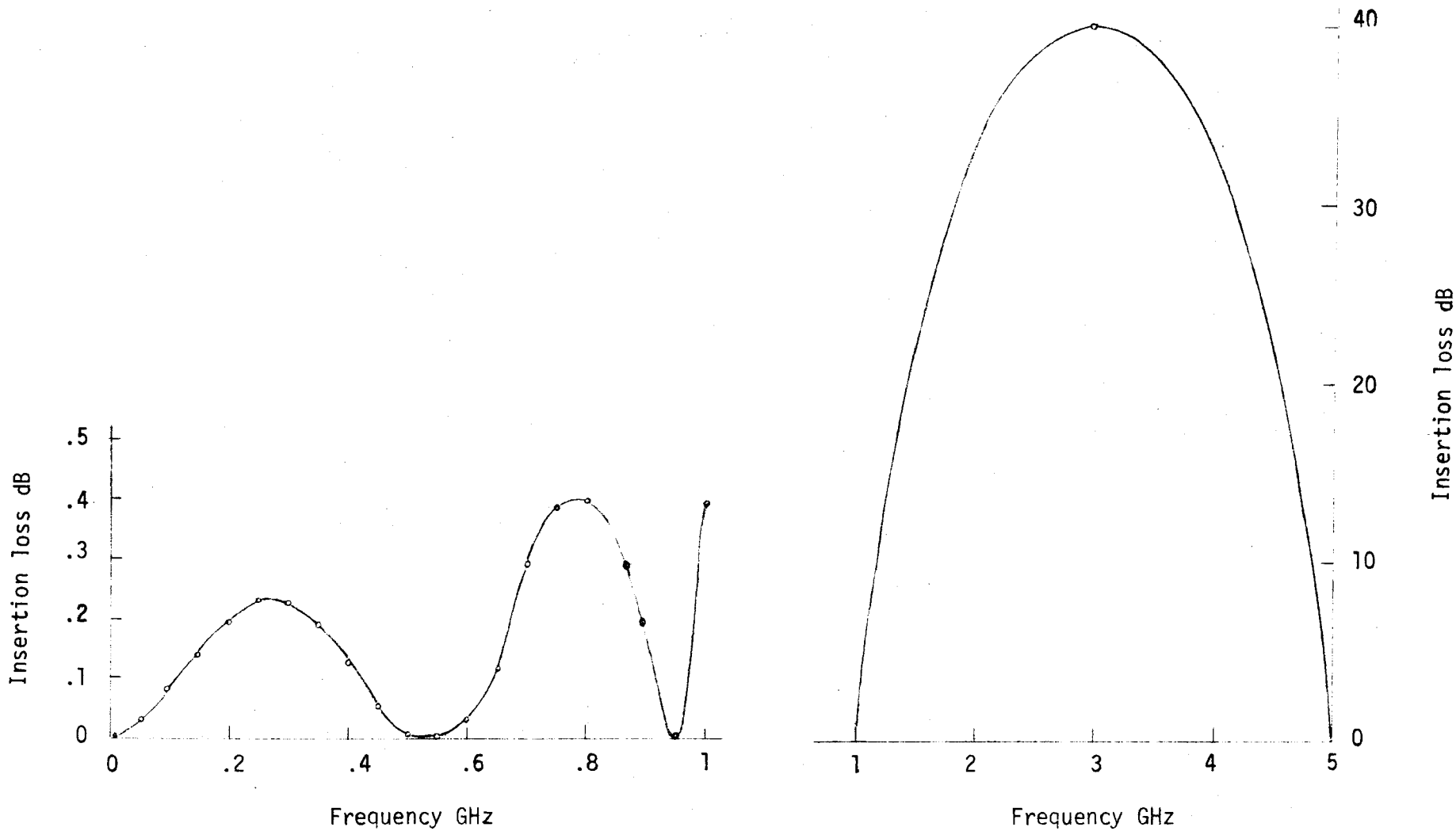


Fig. 6-10 Response of the five section transmission-line filter for the constrained design problem.

CHAPTER VII

Conclusions

Two user-oriented computer programs in FORTRAN IV for the discrete nonlinear least pth approximation [1] and generalized least pth approximation [2], respectively, have been presented in this thesis. The instructions on how to use the packages are given, illustrated with simple examples.

The first program treats the case when we have only one specified function. The package can be used for fitting one function or data to another continuous function when the error function happens to be real. If quadratic interpolation is applied, the specified function and the approximating function have to be continuous on a given interval. Employing quadratic interpolation the sampling for the objective function may take fewer points for equally successful solutions in comparison with the case which does not consider quadratic interpolation, and may save some computation time. Using quadratic interpolation the location of the extreme points were found more precisely and the solutions are closer to the minimax solutions for a given value of p. The computer program can be rearranged such that the specified function and the approximating function are both complex, which is useful in electrical engineering design problems.

The second program is an extension of the first program and is applicable to design problems with upper and lower specifications. The program is directly applicable to such problems as meeting or exceeding design specifications on several disjoint closed intervals as in filter design. Although the program is not written for non-linear programming we found that it is also applicable for problems with parameter constraints.

The methods presented abandon the linear programming subproblem which many of the minimax methods do. The advantage over the direct minimax methods is that they use very efficient gradient methods such as Fletcher-Powell [18] and a recent method by Fletcher [19]. From the experimental results, the Fletcher-Powell algorithm was found to be reliable. The method, however, was found to be slow in comparison with the method proposed by Fletcher. The latter method requires fewer function evaluations to reach the optimum and is less time consuming.

The larger the value of **p** that is used, the more nearly the minimax solution is obtained, but more function evaluations are required to bring the objective function close to the optimum. For practical purposes smaller values of p may be used to attain a satisfactory solution, hence the objective function will be minimized faster. We can start with a smaller value of p, increase it after each complete optimization and terminate when the relative change in the objective function in the successive iterations is less than a prescribed small quantity. This can be a disadvantage if the starting point is close to the minimax optimum, which rarely happens in practice.

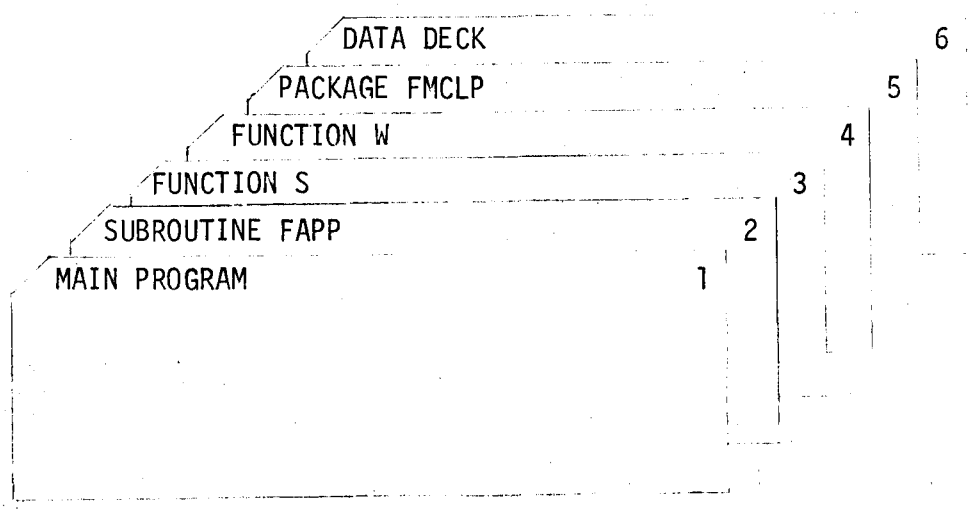
Typically less than 3 minutes of CDC 6400 computer time and a core requirement of about $14K_{10}$ is sufficient to optimize the type of examples given in this thesis to a high degree of accuracy from a good starting point.

APPENDIX I

Usage of FMCLP (Function Minimization for Continuous Least pth Approximation)

Purpose To minimize the objective function of k variables $\underset{\sim}{a}$ defined as a discrete least pth objective with a single specification (5.3) using gradient methods.

How to use Set the input deck as follows:



1. Main program

Write the main program as indicated below:

(i) Dimension the following arrays

A(K), ASTRT(K), G(K), GRAD(K), Y(K), PY(K), DUM1(K), DUM2(K), EPS(K), H(M), X(N2), ERROR(N2),IPA(ITER)

where

K is the number of variable parameters,

M = K(K+7)/2, N2 = N+2,

N is the discrete point set,

ITER is the maximum number of times the optimization method is used.

(ii) Call the subroutine FMCLP as follows:

CALL FMCLP (A, ASTRT, G, GRAD, Y, PY, DUM1, DUM2, EPS, H, X, ERROR, IPA)

(iii) Add STOP and END cards.

2. Subroutine FAPP

This subroutine calculates the approximating function and its gradients with respect to variable vector $\underset{\sim}{a}$.

Write subroutine FAPP as follows:

```
      SUBROUTINE FAPP (X, K, A, APP, GRAD, INDIC)
      DIMENSION A(1), GRAD(1)
```

where X, K, A and INDIC are input and APP and GRAD are output variables.

```
      GO TO (1, 2), INDIC
C     Compute the value of the approximating function
1     APP = F(A, X)
      RETURN
C     Calculate the values of the gradients of the approximating
      function
2     GRAD(1) = ∂F(A, X)/∂A1
```

$$GRAD(2) = \frac{\partial F(\underset{\sim}{A}, X)}{\partial A_2}$$

.

.

.

$$GRAD(K) = \frac{\partial F(\underset{\sim}{A}, X)}{\partial A_k}$$

RETURN

END

where INDIC may have the values 1 or 2 which indicate whether the approximating function or all the gradients should be calculated, respectively.

3. FUNCTION S

Function S is a subprogram of a single input variable X and defines a specified function

$$S = S(X).$$

Add RETURN and END cards.

4. FUNCTION W

Function W is also a subprogram of a single input variable X and defines a weighting function

$$W = w(X).$$

Add RETURN and END cards.

An example which shows how to set these subprograms is the same as Test 3 in the Chapter V. The listings which define this problem follow.

```
      SUBROUTINE FAPP(X,N1,A,APP,GRAD,INDIC)
C
C          SUBROUTINE WHICH CALCULATES APPROXIMATING
C          FUNCTION AND ITS GRADIENTS WITH RESPECT TO
C          VARIABLE PARAMETERS
C
      DIMENSION A(1),GRAD(1)
      GO TO(100,200),INDIC
  100 APP=A(1)*X+A(2)*EXP(X)
      RETURN
  200 GRAD(1)=X
      GRAD(2)=EXP(X)
      RETURN
      END
```

.....

```
      FUNCTION S (X)
C
C          FUNCTION SUBPROGRAM WHICH DEFINES
C          SPECIFIED FUNCTION
C
      S=X**2
      RETURN
      END
```

.....

```
      FUNCTION W(X)
C
C          FUNCTION SUBPROGRAM WHICH DEFINES
C          WEIGHTING FUNCTION
C
      W=1.
      RETURN
      END
```

6. Data Deck

Parameters to be supplied as data are defined below:

N	The number of sample points forming the discrete point set.
XA, XB	The left and the right end points of the interval of the independent parameter.
NSUB	The number of subintervals over [XA, XB].
IREAD	Integer which denotes whether or not the discrete set of points in [XA, XB] will be read. If IREAD=0, the discrete point set will be arranged equidistantly over the interval; If IREAD=1, the discrete point set will be read from data.
X(I), I=1, N	The discrete point set over the interval.
K	The number of the independent variable parameters a_i.
ASTRT(I), I=1, K	Starting values for the K variable parameters.
IGRDCH	Gradients to be checked if IGRDCH=1, it should be set to 0 if gradients are not to be checked.
MET	Optimization method to be called: if MET=1, Fletcher method will be called; if MET=2, Fletcher-Powell method will be called.
MAX	Maximum number of permissible iterations.

ITER	Has already been defined in the main program as a length of the working array.
IPA(I), I=1, ITER	Vector containing the values of p for different least pth objectives.
IOPT	Denotes how many times the optimization is repeated with different starting points and/or different optimization techniques.
IPRINT	Intermediate output is printed out every IPRINT iteration; it should be set to 0 if no intermediate output is desired.
IDATA	Input data is printed out if IDATA=1; it should be set to 0 if input data is not to be printed out.
EST	Minimum estimated value of the objective function.
EPS(I), I=1, K	Small test quantities used by the Fletcher method.
EPS1	Small test quantity used by the Fletcher-Powell method.
DIF	Small test quantity used by the subroutine FMCLP.

Setting up the data deck

	Conditions	Number of cards	Parameters	Type	Format
	-	1	K, N, NSUB, IREAD, IOPT, ITER, IGRDCH	INTEGER	7I10
	-	1	XA, XB	REAL	2E16.8
	IREAD = 1	As many as required by N	X(I), I = 1, N	REAL	5E16.8
	-	1	EST, DIF	REAL	2E16.8
↑	-	As many as required by K	ASTRT(I), I = 1, K	REAL	5E16.8
	-	1	MET, MAX, IPRINT, IDATA	INTEGER	4I10
+ IOPT times	MET = 1	As many as required by K	EPS(I), I=1, K	REAL	5E16.8
	MET = 2	1	EPS1	REAL	5E16.8
↓	-	As many as required by ITER	IPA(I), I = 1, ITER	INTEGER	8I10

Recommended values for some of the parameters

NSUB = 5

MAX = 100

EPS(I), I = 1, K, each 10^{-6}

DIF = 10^{-4}

EST A lower bound of the minimum of the objective function may be obtained from physical reasons. If the true minimum is not known, choose EST to be small enough (negative values are allowed). For approximation problems 0 is convenient.

The following list illustrates how to set the data for the example for which listings have been shown. Both optimization methods, Fletcher-Powell and Fletcher, are called.

```
   2          10          10          0          2          4          1
0.0E    00          2.0E    00
0.0E    00          1.0E    -4
1.0E    00          1.0E    00
   1          50          1           1
1.0E    -6          1.0E    -6
   2          10          100         1000
1.0E    00          1.0E    00
   2          50          1           1
1.0E    -6
   2          10          100         1000
```

APPENDIX II

Listing of the Package FMCLP

```
      SUBROUTINE FMCLP (A,XSTRT,G,GRAD,Y,PY,DUM1,DUM2,EPS,H,X,ERROR,IPA)
C
CC       SUBROUTINE WHICH COORDINATES THE OTHER
CC       SUBROUTINES IN THE PACKAGE FMCLP
C
      EXTERNAL FUNCT
      EXTERNAL S,W
      DIMENSION A(1), XSTRT(1), G(1), Y(1), PY(1), DUM1(1), DUM2(1), EPS
     1(1), H(1), GRAD(1), X(1), ERROR(1), IPA(1)
      COMMON /BLK/ KO
      LOGICAL CONV,UNITH
      EPR(Z)=WERR(Z,G,H,A,GRAD)
      UNITH=.TRUE.
      READ (5,40) N1,N,NPOD,IREAD,IOPT,ITER,IGROCH
      READ (5,42) XA,XB
      XAB=XB-XA
      NP=N+1
      IF (IREAD.EQ.0) IREAD=2
      GO TO (3,1), IREAD
1     X(2)=XA
      K=N-1
      DELTA=XAB/K
      DO 2 I=1,K
      X(I+2)=XA+I*DELTA
2     CONTINUE
      X(N+1)=XB
      GO TO 4
3     READ (5,42) (X(I),I=2,NP)
4     WRITE (6,50)
      READ (5,42) EST,CIF
      READ (5,42) (XSTRT(I),I=1,N1)
      DO 5 I=1,N1
      A(I)=XSTRT(I)
5     CONTINUE
      DO 6 I=2,NP
      II=I-1
      ERROR(I)=ERR(X(I))
      WRITE (6,51) II,X(I),ERROR(I)
6     CONTINUE
      IGROCH=1
      CALL NEWSET (XA,XB,NPOD,S,W,A,GRAD,N,X,ERROR,IQI)
      WRITE (6,43) IQI
      IF (IQI) 9,9,7
7     WRITE (6,45)
      DO 8 I=2,NP
      II=I-1
      ERROR(I)=ERR(X(I))
      WRITE (6,51) II,X(I),ERROR(I)
8     CONTINUE
9     ERROR(1)=ERROR(2)
      DO 10 I=3,NP
      ERROR(1)=AMAX1(ERROR(1),ERROR(I))
10    CONTINUE
      WRITE (6,48)
      WRITE (6,49)
      WRITE (6,47) ERROR(1)
C
CC        DATA FOR THE OPTIMALITY
CC        FOR THE OPTIMIZATION METHOD USED
C
      DO 39 K=1,IOPT
      KP=1
      IF (K-1) 12,12,11
11    READ (5,42) (XSTRT(I),I=1,N1)
12    READ (5,40) MET,MAX,IPRINT,IDATA
      IF (MET.EQ.1) READ (5,42) (EPS(I),I=1,N1)
      IF (MET.EQ.2) READ (5,42) EPS1
      READ (5,40) (IPA(I),I=1,ITER)
      DO 38 KK=1,ITER
C
CC        OPTIMIZATION
C
      IP=IPA(KK)
      IF (KK.GT.1) FF=F
      IF (KP.EQ.0) GO TO 13
      DO 14 I=1,N1
      A(I)=XSTRT(I)
14    CONTINUE
      IF (IGROCH.NE.1) GO TO 15
      CALL GPDCHK (N1,A,G,PY,Y,GRAD,N,X,ERROR,IP,DUM1)
15    IF (KP.EQ.0) GO TO 16
      IF (IDATA.EQ.0) GO TO 16
      M=2
      CALL INPUT (MET,M,MAX,N1,IPRINT,IDATA,EPS1,EST,EPS,XSTRT)
16    IF (MET.EQ.0) MET=4
      INDEX=0
      GO TO (17,22,31,27), MET
17    IF (IPRINT.EQ.1) GO TO 18
      CALL WRITE1 (1)
18    CALL SECOND (T1)
      IF (KP.NE.0) GO TO 20
      DO 19 I=1,N1
      A(I)=DUM1(I)
19    CONTINUE
20    CALL FMNFC (N1,A,F,G,H,UNITH,EST,EPS,MAX,IPRINT,IEXIT,GRAD,N,X,ERR
     1OR,IP)
```

```
      DO 21 I=1,N1                                                          A   97
      DUM1(I)=A(I)                                                          A   98
21    CONTINUE                                                              A   99
      CALL SECOND (T2)                                                      A  100
      CALL FINAL (A,F,N1)                                                   A  101
      T=T2-T1                                                               A  102
      WRITE (6,41) T                                                        A  103
      GO TO 27                                                              A  104
22    IF (IPRINT.EQ.0) GO TO 23                                             A  105
      CALL WRITE1 (2)                                                       A  106
23    CALL SECOND (T1)                                                      A  107
      IF (KR.NE.0) GO TO 25                                                 A  108
      DO 24 I=1,N1                                                          A  109
      A(I)=DUM2(I)                                                          A  110
24    CONTINUE                                                              A  111
25    CALL FMFPC (FUNCT,N1,A,F,G,EST,EPS1,MAX,IER,H,IPRINT,GRAD,N,X,ERRO    A  112
     1R,IP)                                                                 A  113
      DO 26 I=1,N1                                                          A  114
      DUM2(I)=A(I)                                                          A  115
26    CONTINUE                                                              A  116
      CALL SECOND (T2)                                                      A  117
      CALL FINAL (A,F,N1)                                                   A  118
      T=T2-T1                                                               A  119
      WRITE (6,41) T                                                        A  120
27    INDEX=INDEX+1                                                         A  121
      IF (M.EQ.1) GO TO 28                                                  A  122
      GO TO 30                                                              A  123
28    DO 29 I=1,N1                                                          A  124
      A(I)=XSTRT(I)                                                         A  125
29    CONTINUE                                                              A  126
30    CONTINUE                                                              A  127
C                                                                           A  128
31    KR=0                                                                  A  129
      WRITE (6,44) IP                                                       A  130
      WRITE (6,46)                                                          A  131
      DO 32 I=2,NP                                                          A  132
      II=I-1                                                                A  133
      WRITE (6,51) II,X(I),ERROR(I)                                         A  134
32    CONTINUE                                                              A  135
      CALL NEWSET (XA,X8,NPOD,S,W,A,GRAD,N,X,ERROR,IQI)                     A  136
      WRITE (6,43) IQI                                                      A  137
      IF (IQI) 35,35,33                                                     A  138
33    WRITE (6,45)                                                          A  139
      DO 34 I=2,NP                                                          A  140
      II=I-1                                                                A  141
      ERROR(I)=ERR(X(I))                                                    A  142
      WRITE (6,51) II,X(I),ERROR(I)                                         A  143
34    CONTINUE                                                              A  144
35    ERROR(1)=ERROR(2)                                                     A  145
      DO 36 I=3,NP                                                          A  146
      ERROR(1)=AMAX1(ERROR(1),ERROR(I))                                     A  147
36    CONTINUE                                                              A  148
      WRITE (6,48)                                                          A  149
      WRITE (6,49)                                                          A  150
      WRITE (6,47) ERROR(1)                                                 A  151
      IGROCH=IGROCH+2                                                       A  152
      IF (KK-1) 38,38,37                                                    A  153
37    FTST=ABS((FF-F)/FF)                                                   A  154
      IF (FTST.LT.DIF) GO TO 39                                             A  155
38    CONTINUE                                                              A  156
39    CONTINUE                                                              A  157
      RETURN                                                                A  158
C                                                                           A  159
40    FORMAT (8I10)                                                         A  160
41    FORMAT (1HO,//25X,*EXECUTION TIME IN SECONDS=*,F10.5)                 A  161
42    FORMAT (5E16.8)                                                       A  162
43    FORMAT (1H1,14X,*NUMBER OF Q.I.=*,I3)                                 A  163
44    FORMAT (1H1,13X,*P  =*,I7)                                            A  164
45    FORMAT (//8X,*NEW SET OF INDEPENDENT VARIABLE*,15X,*ERRORS*//)        A  165
46    FORMAT (//13X,*INDEPENDENT VARIABLE*,15X,*ERRORS*//)                  A  166
47    FORMAT (13X,E20.12)                                                   A  167
48    FORMAT (//15X,*ABSOLUTE VALUE*)                                       A  168
49    FORMAT (15X,*OF MAXIMUM ERROR*//)                                     A  169
50    FORMAT (1H1,8X,*INITIAL SET OF INDEPENDENT VARIABLE*,9X,*ERRORS*//)   A  170
51    FORMAT (I9,1X,E23.12,10X,E23.12)                                      A  171
      END                                                                   A  172-
```

```
      FUNCTION WERR (Z,S,W,B,GRAD)                                          B    1
C                                                                           B    2
C         FUNCTION SUBPROGRAM WHICH CALCULATES                              B    3
C         WEIGHTED ERROR FUNCTION                                           B    4
C                                                                           B    5
      EXTERNAL S,W                                                          B    6
      DIMENSION B(1), GRAD(1)                                               B    7
      CALL FAPP (Z,N1,B,APP,GRAD,1)                                         B    8
      WERR=(APP-S(Z))*W(Z)                                                  B    9
      RETURN                                                                B   10
      END                                                                   B   11-
```

```
      SUBROUTINE NEWSET (XA,XB,NPOD,S,W,B,GRAD,N,X,ERROR,IQI)                  C    1
C                                                                              C    2
C         SUBROUTINE WHICH CALCULATES THE NEW SET OF THE INDEPENDENT           C    3
C         VARIABLES WHICH INCLUDE ALL THE EXTREMA OF THE WEIGHTED              C    4
C         ERROR FUNCTION                                                       C    5
C                                                                              C    6
      EXTERNAL S,W                                                             C    7
      DIMENSION B(1), GRAD(1), X(1), ERROR(1)                                  C    8
      EPR(Z)=WERR(Z,S,W,B,GRAD)                                                C    9
      EPSN(Z)=ABS(WERR(Z,S,W,B,GRAD))                                          C   10
      IER=0                                                                    C   11
      NN=N+1                                                                   C   12
      NNN=N+2                                                                  C   13
      X(1)=XA                                                                  C   14
      X(NNN)=XB                                                                C   15
      II=2                                                                     C   16
      IQ=1                                                                     C   17
      IND=1                                                                    C   18
      IQI=0                                                                    C   19
      DO 29 I=1,NN                                                             C   20
      IF (X(I)-X(I+1)) 1,29,29                                                 C   21
1     ZMIN=X(I+1)                                                              C   22
      ZMAX=X(I)                                                                C   23
      IQ=IND                                                                   C   24
      IND=1                                                                    C   25
      Z=ZMAX                                                                   C   26
      EMAX=EPSN(Z)                                                             C   27
      EMIN=EPSN(ZMIN)                                                          C   28
      DELTA=(ZMIN-ZMAX)/NPOD                                                   C   29
      NPOD1=NPOD+1                                                             C   30
      MM=1                                                                     C   31
      MIN=NPOD1                                                                C   32
      DO 7 K=1,NPOD                                                            C   33
      ETREN=EPSN(Z)                                                            C   34
      IF (ETREN-EMAX) 6,6,2                                                    C   35
2     EMAX=ETREN                                                               C   36
      MM=K                                                                     C   37
      ZMAX=Z                                                                   C   38
      ZTEST1=ZMAX+DELTA                                                        C   39
      ETEST1=EPSN(ZTEST1)                                                      C   40
      IF (EMAX-ETEST1) 6,6,3                                                   C   41
3     ZTEST2=ZMAX-DELTA                                                        C   42
      IF (XA-(ZMAX-DELTA)) 4,4,6                                               C   43
4     ETEST2=EPSN(ZTEST2)                                                      C   44
      IF (EMAX-ETEST2) 6,6,5                                                   C   45
5     IND=2                                                                    C   46
      GO TO 8                                                                  C   47
6     Z=Z+DELTA                                                                C   48
7     CONTINUE                                                                 C   49
C                                                                              C   50
8     IF (XA-(ZMAX-DELTA)) 9,9,13                                              C   51
9     IF (MM-NPOD) 10,10,13                                                    C   52
10    GO TO (13,11), IND                                                       C   53
11    Q1=EPSN(ZMAX-DELTA)                                                      C   54
      Q2=EPSN(ZMAX+DELTA)                                                      C   55
      RA=(EMAX*2.-Q1-Q2)*2.                                                    C   56
      IF (RA) 12,13,12                                                         C   57
12    ZMAX=ZMAX+(Q2-Q1)*DELTA/RA                                               C   58
      IQI=IQI+1                                                                C   59
      ZMIN=ZTEST1                                                              C   60
C                                                                              C   61
13    IF (I-1) 14,14,17                                                        C   62
14    IF (ZMAX-ZMIN) 15,29,29                                                  C   63
15    GO TO (29,16), IND                                                       C   64
16    X(1)=ZMAX                                                                C   65
      XKP=X(2)                                                                 C   66
      X(2)=ZMIN                                                                C   67
      GO TO 29                                                                 C   68
C                                                                              C   69
17    IF (X(I-1)-ZMAX) 18,21,21                                                C   70
18    IF (ZMAX-ZMIN) 19,21,21                                                  C   71
19    GO TO (23,20), IND                                                       C   72
20    X(I)=ZMAX                                                                C   73
21    IF (X(I)-ZMIN) 22,25,25                                                  C   74
22    XKP=X(I+1)                                                               C   75
      X(I+1)=ZMIN                                                              C   76
      GO TO 23                                                                 C   77
23    GO TO (25,24), IQ                                                        C   78
24    X(I)=XKP                                                                 C   79
C                                                                              C   80
25    IF (I-NN) 29,26,29                                                       C   81
26    IF (X(I)-X(I+1)) 27,29,29                                                C   82
27    IF (EPSN(X(I))-EPSN(X(I+1))) 28,29,29                                    C   83
28    II=II-1                                                                  C   84
29    CONTINUE                                                                 C   85
      IF (EPSN(X(2))-EPSN(X(1))) 30,31,31                                      C   86
30    II=II-3                                                                  C   87
C                                                                              C   88
31    IF (II-2) 32,50,32                                                       C   89
C                                                                              C   90
32    IF (II+2) 42,33,42                                                       C   91
33    JJ=NN/2+1                                                                C   92
      IF (N-2) 39,39,34                                                        C   93
```

```
      IF (EPSN(X(JJ-1))-EPSN(X(JJ+1))) 35,36,36                          C   94
35    JJ=JJ-1                                                            C   95
      GO TO 37                                                           C   96
36    JJ=JJ+1                                                            C   97
37    IF (EPSN(X(JJ-1))-EPSN(X(JJ+1))) 38,39,39                          C   98
38    JJ=JJ-1                                                            C   99
39    DO 40 K=2,JJ                                                       C  100
      I=JJ+2-K                                                           C  101
      X(I)=X(I-1)                                                        C  102
40    CONTINUE                                                           C  103
      JJJ=JJ+1                                                           C  104
      DO 41 K=JJJ,NN                                                     C  105
      I=K+1                                                              C  106
      X(I-1)=X(I)                                                        C  107
41    CONTINUE                                                           C  108
C                                                                        C  109
42    IF (II-1) 46,43,46                                                 C  110
43    IF (EPSN(X(NNN))-EPSN(X(2))) 50,44,44                              C  111
C                                                                        C  112
44    DO 45 I=2,NN                                                       C  113
      X(I)=X(I+1)                                                        C  114
45    CONTINUE                                                           C  115
46    IF (II+1) 50,47,50                                                 C  116
47    IF (EPSN(X(1))-EPSN(X(NN))) 50,50,48                               C  117
C                                                                        C  118
48    DO 49 K=2,NN                                                       C  119
      I=NN+2-K                                                           C  120
      X(I)=X(I-1)                                                        C  121
49    CONTINUE                                                           C  122
50    CONTINUE                                                           C  123
      RETURN                                                             C  124
      END                                                                C  125-
```

```
      SUBROUTINE FUNCT (N1,B,OBJ,G,GRAD,N,X,ERROR,IP)                    D    1
C                                                                        D    2
C         SUBROUTINE WHICH SELECTS THE MAXIMUM ERROR                     D    3
C         AND COMPUTES THE OBJECTIVE FUNCTION AND ITS                    D    4
C         GRADIENTS W.R.T. THE VARIABLE PARAMETERS                       D    5
C         IN THE LEAST P-TH SENSE                                        D    6
C                                                                        D    7
      EXTERNAL S,W                                                       D    8
      DIMENSION B(1), G(1), GRAD(1), X(1), ERROR(1)                      D    9
      ERR(Z)=WERR(Z,S,W,B,GRAD)                                          D   10
      OBJP=0.                                                            D   11
      GRADP=0.                                                           D   12
      DO 1 K=1,N1                                                        D   13
      G(K)=0.                                                            D   14
1     CONTINUE                                                           D   15
      NN=N+1                                                             D   16
      ERROR(1)=0.                                                        D   17
      DO 2 I=2,NN                                                        D   18
      ERROR(I)=ERR(X(I))                                                 D   19
2     CONTINUE                                                           D   20
      DO 4 I=2,NN                                                        D   21
      IF (ABS(ERROR(I))-ERROR(1)) 4,4,3                                  D   22
3     ERROR(1)=ABS(ERROR(I))                                             D   23
4     CONTINUE                                                           D   24
      DO 6 I=2,NN                                                        D   25
      Z=X(I)                                                             D   26
      DEC=ERROR(I)/ERROR(1)                                              D   27
      DEL=ABS(DEC)                                                       D   28
      OBJI=DEL**IP                                                       D   29
      GRADI=DEL**(IP-2)*DEC                                              D   30
      OBJP=OBJP+OBJI                                                     D   31
      CALL FAPP (Z,N1,B,APP,GRAD,2)                                      D   32
      DO 5 K=1,N1                                                        D   33
      GRAD(K)=GRADI*W(Z)*GRAD(K)                                         D   34
      G(K)=G(K)+GRAD(K)                                                  D   35
5     CONTINUE                                                           D   36
6     CONTINUE                                                           D   37
      PP=1./IP                                                           D   38
      OBJ=ERROR(1)*(OBJP**PP)                                            D   39
      GRP=OBJP**(PR-1.)                                                  D   40
      DO 7 K=1,N1                                                        D   41
      G(K)=GRP*G(K)                                                      D   42
7     CONTINUE                                                           D   43
      RETURN                                                             D   44
      END                                                                D   45-
```

```
      SUBROUTINE GRDCHK (N,X,G,PY,Y,GRAD,NP,XP,ERROR,IP,DUM1)            E    1
C                                                                        E    2
C         SUBROUTINE WHICH CHECKS THE GRADIENTS                          E    3
C         W.R.T. ALL VARIABLE PARAMETERS                                 E    4
C                                                                        E    5
      DIMENSION X(1), G(1), PY(1), Y(1), GRAD(1), XP(1), ERROR(1), DUM1( E    6
     11)                                                                 E    7
```

```
      CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)                          E    8
      DO 1 I=1,N                                                        E    9
      DELX=1.E-4*X(I)                                                   E   10
      X(I)=X(I)+DELX                                                    E   11
      CALL FUNCT (N,X,FNEW,PY,GRAD,NP,XP,ERROR,IP)                      E   12
      Y(I)=(FNEW-F)/DELX                                                E   13
      DUM1(I)=Y(I)                                                      E   14
      X(I)=X(I)-DELX                                                    E   15
1     CONTINUE                                                          E   16
      DO 2 I=1,N                                                        E   17
      IF (ABS(Y(I)).LT.1.E-20) DUM1(I)=1.E-20                           E   18
      PY(I)=ABS((Y(I)-G(I))/DUM1(I))*100.                               E   19
2     CONTINUE                                                          E   20
      WRITE (6,6)                                                       E   21
      WRITE (6,7)                                                       E   22
      WRITE (6,8) (I,X(I),I=1,N)                                        E   23
      WRITE (6,9)                                                       E   24
      DO 3 I=1,N                                                        E   25
      WRITE (6,10) G(I),Y(I),PY(I)                                      E   26
3     CONTINUE                                                          E   27
      DO 4 I=1,N                                                        E   28
      IF (PY(I).GT.10.) GO TO 5                                         E   29
4     CONTINUE                                                          E   30
      WRITE (6,11)                                                      E   31
      RETURN                                                            E   32
5     WRITE (6,12)                                                      E   33
      CALL EXIT                                                         E   34
C                                                                       E   35
6     FORMAT (1H1)                                                      E   36
7     FORMAT (1HO,5X,*GRADIENTS CHECKING*,/,6X,18(*-*),//,6X,*GRADIENTS E   37
     1HAVE BEEN CHECKED AT THE FOLLOWING POINT*/)                       E   38
8     FORMAT (10X,*X(*,I2,*)=*,E16.8)                                   E   39
9     FORMAT (///,1HO,5X,*ANALYTICAL GRADIENTS*,5X,*NUMERICAL GRADIENTS* E   40
     1,5X,*PERCENTAGE ERROR*,/)                                         E   41
10    FORMAT (1HO,5X,3(E16.8,9X))                                       E   42
11    FORMAT (1HO,//,6X,*GRADIENTS ARE O. K.*)                          E   43
12    FORMAT (1HO,//,6X,*YOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRADIE E   44
     1NTS ARE INCORRECT*,/6X,*PLEASE CHECK IT AGAIN*)                   E   45
      END                                                               E   46-
```

.....

```
      SUBROUTINE FMNFC (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT,GRAD F    1
     1,NP,XP,ERROR,IP)                                                  F    2
C                                                                       F    3
C        PURPOSE                                                        F    4
C           TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES  F    5
C           ASSUMING THAT ITS GRADIENTS CAN BE CALCULATED EXPLICITLY    F    6
C           BY THE METHOD OF FLETCHER                                   F    7
C                                                                       F    8
C           THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE            F    9
C           R. FLETCHER, A NEW APPROACH TO VARIABLE METRIC ALGORITHMS,  F   10
C           COMP. JOURNAL, VOL.13, 1970, PP.317-322.                    F   11
C                                                                       F   12
      DIMENSION X(1), G(1), H(1), EPS(1), GRAD(1), XP(1), ERROR(1)      F   13
      LOGICAL CONV,UNITH                                                F   14
      COMMON /BLK/ KO                                                   F   15
      CALL SECOND (T3)                                                  F   16
      KO=0                                                              F   17
      CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)                          F   18
      IF (F.LT.FEST) GO TO 23                                           F   19
      NFNS=1                                                            F   20
      ITN=0                                                             F   21
      STEP=1.                                                           F   22
      IDX=N                                                             F   23
      IDG=N+N                                                           F   24
      IH=IDG+N                                                          F   25
      IF (.NOT.UNITH) GO TO 2                                           F   26
      IJ=IH+1                                                           F   27
      DO 1 I=1,N                                                        F   28
      DO 1 J=I,N                                                        F   29
      H(IJ)=0.                                                          F   30
      IF (I.EQ.J) H(IJ)=1.0                                             F   31
1     IJ=IJ+1                                                           F   32
2     CONV=.TRUE.                                                       F   33
      GDX=0.                                                            F   34
      DO 6 I=1,N                                                        F   35
      Z=0.                                                              F   36
      IJ=IH+I                                                           F   37
      IF (I.EQ.1) GO TO 4                                               F   38
      II=I-1                                                            F   39
      DO 3 J=1,II                                                       F   40
      Z=Z-H(IJ)*G(J)                                                    F   41
      IJ=IJ+N-J                                                         F   42
3     CONTINUE                                                          F   43
4     DO 5 J=I,N                                                        F   44
      Z=Z-H(IJ)*G(J)                                                    F   45
      IJ=IJ+1                                                           F   46
5     CONTINUE                                                          F   47
      IF (ABS(Z).GT.EPS(I)) CONV=.FALSE.                                F   48
      H(IDX+I)=Z                                                        F   49
      GDX=GDX+G(I)*Z                                                    F   50
6     CONTINUE                                                          F   51
```

```
C
      IF (IPRINT.EQ.0) GO TO 7                                          F   52
      IF (MOD(ITN,IPRINT).NE.0) GO TO 7                                 F   53
      CALL SECOND (T4)                                                  F   54
      TIME=T4-T3                                                        F   55
      CALL WRITE2 (X,N,G,F,NFNS,ITN,TIME)                               F   56
7     IEXIT=1                                                           F   57
      IF (CONV) GO TO 24                                                F   58
      IEXIT=2                                                           F   59
      IF (GDX.GE.0.) GO TO 24                                           F   60
      Z=1.                                                              F   61
      IF (ITN.LT.N.AND.UNITH) Z=STEP                                    F   62
      W=2.*(FEST-F)/GDX                                                 F   63
      IF (W.LT.Z) Z=W                                                   F   64
      STEP=Z                                                            F   65
8     GDX=GDX*Z                                                         F   66
      DO 9 I=1,N                                                        F   67
      H(IDX+I)=H(IDX+I)*Z                                               F   68
      X(I)=X(I)+H(IDX+I)                                                F   69
9     CONTINUE                                                          F   70
      CALL FUNCT (N,X,FP,H,GRAD,NP,XP,ERROR,IP)                         F   71
      IF (FP.LT.FEST) GO TO 23                                          F   72
      NFNS=NFNS+1                                                       F   73
      IEXIT=3                                                           F   74
      IF (ITN.EQ.MAXFN) GO TO 24                                        F   75
      GPDX=0.                                                           F   76
      DO 10 I=1,N                                                       F   77
      H(IDG+I)=H(I)-G(I)                                                F   78
      GPDX=GPDX+H(I)*H(IDX+I)                                           F   79
10    CONTINUE                                                          F   80
      DGDX=GPDX-GDX                                                     F   81
      IF (F.GT.FP-.0001*GDX) GO TO 12                                   F   82
      IEXIT=4                                                           F   83
      IF (GPDX.LT.0..AND.ITN.GT.N) GO TO 24                             F   84
      Z=3.*(F-FP)+GPDX+GDX                                              F   85
      W=SQRT(1.-GDX/Z*GPDX/Z)*ABS(Z)                                    F   86
      Z=1.-(GPDX+W-Z)/(DGDX+2.*W)                                       F   87
      IF (Z.LT.0.1) Z=0.1                                               F   88
      DO 11 I=1,N                                                       F   89
      X(I)=X(I)-H(IDX+I)                                                F   90
11    CONTINUE                                                          F   91
      GO TO 14                                                          F   92
12    F=FP                                                              F   93
      DO 13 I=1,N                                                       F   94
      G(I)=H(I)                                                         F   95
13    CONTINUE                                                          F   96
      IF (DGDX.GT.0.) GO TO 15                                          F   97
      GDX=GPDX                                                          F   98
      Z=4.                                                              F   99
14    STEP=Z*STEP                                                       F  100
      GO TO 8                                                           F  101
15    IF (GPDX.LT.0.5*GDX) STEP=2.*STEP                                 F  102
      DGHDG=0.                                                          F  103
      DO 19 I=1,N                                                       F  104
      Z=0.                                                              F  105
      IJ=IH+I                                                           F  106
      IF (I.EQ.1) GO TO 17                                              F  107
      II=I-1                                                            F  108
      DO 16 J=1,II                                                      F  109
      Z=Z+H(IJ)*H(IDG+J)                                                F  110
      IJ=IJ+N-J                                                         F  111
16    CONTINUE                                                          F  112
17    DO 18 J=I,N                                                       F  113
      Z=Z+H(IJ)*H(IDG+J)                                                F  114
      IJ=IJ+1                                                           F  115
18    CONTINUE                                                          F  116
      DGHDG=DGHDG+Z*H(IDG+I)                                            F  117
      H(I)=Z                                                            F  118
19    CONTINUE                                                          F  119
      IF (DGHDG.LT.0.0) DGHDG=DGDX*0.01                                 F  120
      IF (DGDX.LT.DGHDG) GO TO 21                                       F  121
      W=1.0+DGHDG/DGDX                                                  F  122
      DO 20 I=1,N                                                       F  123
      H(IDX+I)=W*H(IDX+I)-H(I)                                          F  124
20    CONTINUE                                                          F  125
      DGDX=DGDX+DGHDG                                                   F  126
      DGHDG=DGDX                                                        F  127
21    IJ=IH                                                             F  128
      DO 22 I=1,N                                                       F  129
      W=H(IDX+I)/DGDX                                                   F  130
      Z=H(I)/DGHDG                                                      F  131
      DO 22 J=I,N                                                       F  132
      IJ=IJ+1                                                           F  133
22    H(IJ)=H(IJ)+W*H(IDX+J)-Z*H(J)                                     F  134
      ITN=ITN+1                                                         F  135
      GO TO 2                                                           F  136
23    IEXIT=5                                                           F  137
24    IF (IEXIT.EQ.1) KQ=1                                              F  138
      IF (IPRINT.EQ.0) RETURN                                           F  139
      GO TO (25,26,27,28), IEXIT                                        F  140
25    WRITE (6,30) IEXIT                                                F  141
      GO TO 29                                                          F  142
26    WRITE (6,31) IEXIT                                                F  143
      GO TO 29                                                          F  144
27    WRITE (6,32) IEXIT                                                F  145
      GO TO 29                                                          F  146
                                                                        F  147
```

```
28    WRITE (6,33) IEXIT                                                F 148
29    CONTINUE                                                          F 149
      RETURN                                                            F 150
C                                                                       F 151
30    FORMAT (/,1HO,*IEXIT=*,I2,*CRITERION FOR OPTIMUM (CHANGE IN VECTOR F 152
     1 X .LT.EPS) HAS BEEN SATISFIED*)                                  F 153
31    FORMAT (/,1HO,*IEXIT=*,I2,*EITHER OF THE FOLLOWING THINGS HAS HAPP F 154
     1ENED*,/,9X,*1. EPS CHOSEN IS TOO SMALL*,/,9X,*2. GRADIENTS ARE NOT F 155
     2 CORRECT*,/,9X,*3. MATRIX H GOES SINGULAR*)                       F 156
32    FORMAT (/,1HO,*IEXIT=*,I2,*MAXIMUM NUMBER OF ALLOWABLE ITERATION H F 157
     1AS BEEN EXCEEDED*)                                                F 158
33    FORMAT (/,1HO,*IEXIT=*,I2,*FUNCTION VALUE LESS THAN MINIMUM ESTIMA F 159
     1TED HAS BEEN DETECTED*)                                           F 160
      END                                                               F 161-
```

```
      SUBROUTINE FMFPC (FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H,IPRINT,GRAD,NP  G   1
     1,XP,ERROR,IP)                                                     G   2
C                                                                       G   3
C        PURPOSE                                                        G   4
C           TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES  G   5
C           ASSUMING THAT ITS GRADIENTS CAN BE CALCULATED EXPLICITLY    G   6
C           BY THE METHOD OF FLETCHER AND POWELL                        G   7
C                                                                       G   8
C           THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE            G   9
C           R. FLETCHER AND M.J.D. POWELL, A RAPIDLY CONVERGENT         G  10
C           DESCENT METHOD FOR MINIMIZATION, COMP. JOURNAL,             G  11
C           VOL.6, 1963, PP.163-168.                                    G  12
C                                                                       G  13
      COMMON /BLK/ KO                                                   G  14
C                                                                       G  15
C        DIMENSIONED DUMMY VARIABLES                                    G  16
      DIMENSION H(1), X(1), G(1), GRAD(1), XP(1), ERROR(1)              G  17
C                                                                       G  18
C        COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT G  19
      KO=0                                                              G  20
      CALL SECOND (T3)                                                  G  21
      CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)                          G  22
      KOUNT=0                                                           G  23
      NUMF=1                                                            G  24
      CALL SECOND (T4)                                                  G  25
      TIME=T4-T3                                                        G  26
      IF (IPRINT.EQ.0) GO TO 1                                          G  27
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)                             G  28
1     CONTINUE                                                          G  29
C                                                                       G  30
C        RESET ITERATION COUNTER AND GENERATE IDENTITY MATRIX           G  31
      IER=0                                                             G  32
      KK=0                                                              G  33
      N2=N+N                                                            G  34
      N3=N2+N                                                           G  35
      N3=1=N3+1                                                         G  36
2     K=N31                                                             G  37
      DO 5 J=1,N                                                        G  38
      H(K)=1.                                                           G  39
      NJ=N-J                                                            G  40
      IF (NJ) 6,6,3                                                     G  41
3     DO 4 L=1,NJ                                                       G  42
      KL=K+L                                                            G  43
      H(KL)=0.                                                          G  44
4     CONTINUE                                                          G  45
      K=KL+1                                                            G  46
5     CONTINUE                                                          G  47
C                                                                       G  48
C        START ITERATION LOOP                                           G  49
6     IF (KOUNT.EQ.0) GO TO 7                                           G  50
      IF (KK.NE.IPRINT) GO TO 7                                         G  51
      KK=0                                                              G  52
      CALL SECOND (T4)                                                  G  53
      TIME=T4-T3                                                        G  54
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)                             G  55
7     CONTINUE                                                          G  56
      KOUNT=KOUNT+1                                                     G  57
      KK=KK+1                                                           G  58
C                                                                       G  59
C        SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR       G  60
      OLDF=F                                                            G  61
      DO 11 J=1,N                                                       G  62
      K=N+J                                                             G  63
      H(K)=G(J)                                                         G  64
      K=K+N                                                             G  65
      H(K)=X(J)                                                         G  66
C                                                                       G  67
C        DETERMINE DIRECTION VECTOR H                                   G  68
      K=J+N3                                                            G  69
      T=0.                                                              G  70
      DO 10 L=1,N                                                       G  71
      T=T-G(L)*H(K)                                                     G  72
      IF (L-J) 8,9,9                                                    G  73
8     K=K+N-L                                                           G  74
      GO TO 10                                                          G  75
9     K=K+1                                                             G  76
```

```
10    CONTINUE                                                          G  177
      H(J)=T                                                            G  178
11    CONTINUE                                                          G  179
C                                                                       G  180
C        CHECK WHETHER FUNCTION WILL DECREASE STEPPING ALONG H.         G  181
      DY=0.                                                             G  182
      HNRM=0.                                                           G  183
      GNRM=0.                                                           G  184
C                                                                       G  185
C        CALCULATE DIRECTIONAL DERIVATIVE AND TESTVALUES FOR DIRECTION  G  186
C        VECTOR H AND GRADIENT VECTOR G.                                G  187
      DO 12 J=1,N                                                       G  188
      HNRM=HNRM+ABS(H(J))                                               G  189
      GNRM=GNRM+ABS(G(J))                                               G  190
      DY=DY+H(J)*G(J)                                                   G  191
12    CONTINUE                                                          G  192
C                                                                       G  193
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTIONAL  G  194
C        DERIVATIVE APPEARS TO BE POSITIVE OR ZERO.                     G  195
      IF (DY) 13,57,57                                                  G  196
C                                                                       G  197
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION    G  198
C        VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.               G  199
13    IF (HNRM/GNRM-EPS) 57,57,14                                       G  200
C                                                                       G  201
C        SEARCH MINIMUM ALONG DIRECTION H                               G  202
C                                                                       G  203
C        SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE             G  204
14    FY=F                                                              G  205
      ALFA=2.*(EST-F)/DY                                                G  206
      AMBDA=1.                                                          G  207
C                                                                       G  208
C        USE ESTIMATE FOR STEPSIZE ONLY IF IT IS POSITIVE AND LESS THAN G  209
C        1. OTHERWISE TAKE 1. AS STEPSIZE                               G  210
      IF (ALFA) 17,17,15                                                G  211
15    IF (ALFA-AMBDA) 16,17,17                                          G  212
16    AMBDA=ALFA                                                        G  213
17    ALFA=0.                                                           G  214
C                                                                       G  215
C        SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT           G  216
18    FX=FY                                                             G  217
      DX=DY                                                             G  218
C                                                                       G  219
C        STEP ARGUMENT ALONG H                                          G  220
      DO 19 I=1,N                                                       G  221
      X(I)=X(I)+AMBDA*H(I)                                              G  222
19    CONTINUE                                                          G  223
C                                                                       G  224
C        COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT           G  225
      CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)                          G  226
      NUMF=NUMF+1                                                       G  227
      FY=F                                                              G  228
C                                                                       G  229
C        COMPUTE DIRECTIONAL DERIVATIVE DY FOR NEW ARGUMENT.  TERMINATE G  230
C        SEARCH, IF DY IS POSITIVE. IF DY IS ZERO THE MINIMUM IS FOUND  G  231
      DY=0.                                                             G  232
      DO 20 I=1,N                                                       G  233
      DY=DY+G(I)*H(I)                                                   G  234
20    CONTINUE                                                          G  235
      IF (DY) 21,41,24                                                  G  236
C                                                                       G  237
C        TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT     G  238
C        A MINIMUM HAS BEEN PASSED                                      G  239
21    IF (FY-FX) 22,24,24                                               G  240
C                                                                       G  241
C        REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES         G  242
22    AMBDA=AMBDA+ALFA                                                  G  243
      ALFA=AMBDA                                                        G  244
C        END OF SEARCH LOOP                                             G  245
C                                                                       G  246
C        TERMINATE IF THE CHANGE IN ARGUMENT GETS VERY LARGE            G  247
      IF (HNRM*AMBDA-1.E10) 18,18,23                                    G  248
C                                                                       G  249
C        LINEAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS       G  250
23    IER=2                                                             G  251
      GO TO 62                                                          G  252
C                                                                       G  253
C        INTERPOLATE CUBICALLY IN THE INTERVAL DEFINED BY THE SEARCH    G  254
C        ABOVE AND COMPUTE THE ARGUMENT X FOR WHICH THE INTERPOLATION   G  255
C        POLYNOMIAL IS MINIMIZED                                        G  256
24    T=0.                                                              G  257
25    IF (AMBDA) 26,41,26                                               G  258
26    Z=3.*(FX-FY)/AMBDA+DX+DY                                          G  259
      ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))                                G  260
      DALFA=Z/ALFA                                                      G  261
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA                                 G  262
      IF (DALFA) 57,27,27                                               G  263
27    W=ALFA*SQRT(DALFA)                                                G  264
      ALFA=DY-DX+W+W                                                    G  265
      IF (ALFA) 28,29,28                                                G  266
28    ALFA=(DY-Z+W)/ALFA                                                G  267
      GO TO 30                                                          G  268
29    ALFA=(Z+DY-W)/(Z+DX+Z+DY)                                         G  269
30    ALFA=ALFA*AMBDA                                                   G  270
      DO 31 I=1,N                                                       G  271
      X(I)=X(I)+(T-ALFA)*H(I)                                           G  272
```

```
31    CONTINUE                                                          G 173
C                                                                       G 174
C        TERMINATE, IF THE VALUE OF THE ACTUAL FUNCTION AT X IS LESS    G 175
C        THAN THE FUNCTION VALUES AT THE INTERVAL ENDS, OTHERWISE REDUCE G 176
C        THE INTERVAL BY CHOOSING ONE END-POINT EQUAL TO X AND REPEAT   G 177
C        THE INTERPOLATION.  WHICH END-POINT IS CHOSEN DEPENDS ON THE   G 178
C        VALUE OF THE FUNCTION AND ITS GRADIENT AT X                    G 179
C                                                                       G 180
      NUMF=NUMF+1                                                       G 181
      CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)                          G 182
      IF (F-FX) 32,32,33                                                G 183
32    IF (F-FY) 41,41,33                                                G 184
33    DALFA=0.                                                          G 185
      DO 34 I=1,N                                                       G 186
      DALFA=DALFA+G(I)*H(I)                                             G 187
34    CONTINUE                                                          G 188
      IF (DALFA) 35,38,38                                               G 189
35    IF (F-FX) 37,36,36                                                G 190
36    IF (DX-DALFA) 37,41,37                                            G 191
37    FX=F                                                              G 192
      DX=DALFA                                                          G 193
      T=ALFA                                                            G 194
      AMBDA=ALFA                                                        G 195
      GO TO 25                                                          G 196
38    IF (FY-F) 40,39,40                                                G 197
39    IF (DY-DALFA) 40,41,40                                            G 198
40    FY=F                                                              G 199
      DY=DALFA                                                          G 200
      AMBDA=AMBDA-ALFA                                                  G 201
      GO TO 24                                                          G 202
C                                                                       G 203
C        TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION G 204
41    IF (OLDF-F+EPS) 57,42,42                                          G 205
C                                                                       G 206
C        COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM       G 207
C        TWO CONSECUTIVE ITERATIONS                                     G 208
42    DO 43 J=1,N                                                       G 209
      K=N+J                                                             G 210
      H(K)=G(J)-H(K)                                                    G 211
      K=N+K                                                             G 212
      H(K)=X(J)-H(K)                                                    G 213
43    CONTINUE                                                          G 214
C                                                                       G 215
C        TEST LENGTH OF ARGUMENT DIFFERENCE VECTOR AND DIRECTION VECTOR G 216
C        IF AT LEAST N ITERATIONS HAVE BEEN EXECUTED, TERMINATE, IF     G 217
C        BOTH ARE LESS THAN  EPS                                        G 218
      IER=0                                                             G 219
      IF (KOUNT-N) 47,44,44                                             G 220
44    T=0.                                                              G 221
      Z=0.                                                              G 222
      DO 45 J=1,N                                                       G 223
      K=N+J                                                             G 224
      W=H(K)                                                            G 225
      K=K+N                                                             G 226
      T=T+ABS(H(K))                                                     G 227
      Z=Z+W*H(K)                                                        G 228
45    CONTINUE                                                          G 229
      IF (HNR1-EPS) 46,46,47                                            G 230
46    IF (T-EPS) 62,62,47                                               G 231
C                                                                       G 232
C        TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED  LIMIT         G 233
47    IF (KOUNT-LIMIT) 48,55,55                                         G 234
C                                                                       G 235
C        PREPARE UPDATING OF MATRIX H                                   G 236
48    ALFA=0.                                                           G 237
      DO 52 J=1,N                                                       G 238
      K=J+N3                                                            G 239
      W=0.                                                              G 240
      DO 51 L=1,N                                                       G 241
      KL=N+L                                                            G 242
      W=W+H(KL)*H(K)                                                    G 243
      IF (L-J) 49,50,50                                                 G 244
49    K=K+N-L                                                           G 245
      GO TO 51                                                          G 246
50    K=K+1                                                             G 247
51    CONTINUE                                                          G 248
      K=N+J                                                             G 249
      ALFA=ALFA+W*H(K)                                                  G 250
      H(J)=W                                                            G 251
52    CONTINUE                                                          G 252
C                                                                       G 253
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS      G 254
C        ARE NOT SATISFACTORY                                           G 255
      IF (Z*ALFA) 53,2,53                                               G 256
C                                                                       G 257
C        UPDATE MATRIX H                                                G 258
53    K=N3                                                              G 259
      DO 54 L=1,N                                                       G 260
      KL=N2+L                                                           G 261
      DO 54 J=L,N                                                       G 262
      NJ=N2+J                                                           G 263
      H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA                            G 264
54    K=K+1                                                             G 265
      GO TO 6                                                           G 266
C        END OF ITERATION LOOP                                          G 267
C                                                                       G 268
```

```
C        NO CONVERGENCE AFTER  LIMIT  ITERATIONS                                  G 269
55     IER=1                                                                      GG 270
       IF (KK.NE.IPRINT) GO TO 56                                                 GG 271
       CALL WRITE2 (X,N,G,F,NUMF,KOUNT)                                           GG 272
56     CONTINUE                                                                   GG 273
       GO TO 62                                                                   GG 274
C                                                                                 GG 275
C        RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS                             GG 276
57     DO 58 J=1,N                                                                GG 277
       K=N2+J                                                                     GG 278
       X(J)=H(K)                                                                  GG 279
58     CONTINUE                                                                   GG 280
       CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)                                   GG 281
       NUMF=NUMF+1                                                                GG 282
C                                                                                 GG 283
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE             GG 284
C        FAILS TO BE SUFFICIENTLY SMALL                                           GG 285
       IF (GNRM-EPS) 61,61,59                                                     GG 286
C                                                                                 GG 287
C        TEST FOR REPEATED FAILURE OF ITERATION                                   GG 288
59     IF (IER) 62,60,60                                                          GG 289
60     IER=-1                                                                     GG 290
       GO TO 2                                                                    GG 291
61     IER=0                                                                      GG 292
62     II=IER+2                                                                   GG 293
       IF (II.EQ.2) KK=1                                                          GG 294
       IF (IPRINT.EQ.0) RETURN                                                    GG 295
       GO TO (63,64,65,66), II                                                    GG 296
63     WRITE (6,68) IER                                                           GG 297
       GO TO 67                                                                   GG 298
64     WRITE (6,69) IER                                                           GG 299
       GO TO 67                                                                   GG 300
65     WRITE (6,70) IER                                                           GG 301
       GO TO 67                                                                   GG 302
66     WRITE (6,71) IER                                                           GG 303
67     RETURN                                                                     GG 304
C                                                                                 GG 305
68     FORMAT (1HO,*IER=*,I2,* ERROR IN GRADIENTS CALCULATIONS*)                  GG 306
69     FORMAT (1HO,*IER=*,I2,* CRITERION FOR OPTIMUM HAS BEEN SATISFIED*)         GG 307
70     FORMAT (1HO,*IER=*,I2,* MAXIMUM NUMBER OF ALLOWABLE ITERATIONS HAS         GG 308
     1 BEEN EXCEEDED*)                                                            GG 309
71     FORMAT (1HO,*IER=*,I2,* CHANGE IN ARGUMENTS GETS TOO LARGE, LINEAR         GG 310
     1 SEARCH INDICATES THAT NO MINIMUM EXISTS*)                                  GG 311
       END                                                                        G 312-

HSBZ00L  //// END OF LIST ////
```

APPENDIX III

Listings of the common subroutines for FMCLP and FMLPO packages

Add the subroutines INPUT, FINAL, WRITE1 and WRITE2 to both packages, FMCLP and FMLPO. They are similar to ones presented in the package for function minimization [36] programmed by V. K. Jha.

Listings of the subroutines INPUT, FINAL, WRITE1 and WRITE2 follow.

```
      SUBROUTINE INPUT (MET,M,MAX,N,IPRINT,IDATA,EPS1,EST,EPS,XSTRT)
C
CC        PRINTS THE INPUT DATA
CC        FOR THE OPTIMIZATION PROCESS
C
      DIMENSION XSTRT(1), EPS(1)
      WRITE (6,5)
      IF (MET.NE.1.AND.MET.NE.2) GO TO 4
      INDEX=0
      GO TO (1,2), MET
1     WRITE (6,6)
      GO TO 3
2     WRITE (6,7)
3     CONTINUE
      WRITE (6,8) N
      WRITE (6,9) MAX
      WRITE (6,10) IPRINT
      WRITE (6,11) M
      WRITE (6,12) XSTRT(1)
      WRITE (6,13) (I,XSTRT(I),I=2,N)
      IF (MET.EQ.1) WRITE (6,14) EPS(1)
      IF (MET.EQ.1) WRITE (6,15) (I,EPS(I),I=2,N)
      IF (MET.EQ.2) WRITE (6,16) EPS1
      WRITE (6,17) EST
      RETURN
4     WRITE (6,18)
      CALL EXIT
C
5     FORMAT (1H1,*INPUT DATA*,/,1X,10(*-*),//,1X,*FOLLOWING METHODS HAV
     1E BEEN CALLED*,/)
6     FORMAT (1HO,*NEW FLETCHER METHOD*)
7     FORMAT (1HO,*FLETCHER-POWELL METHOD*)
8     FORMAT (1HO,/1X,*NUMBER OF INDEPENDENT VARIABLES*,36(*.*),*N=*,I5,
     1/)
9     FORMAT (1HO,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS*,27(*.*),*MAX=
     1*,I5,/)
10    FORMAT (1HO,*INTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITERAT
     1IONS*,5(*.*),*IPRINT=*,I5,/)
11    FORMAT (1HO,*STARTING POINT TO BE SAME FOR ALL THE METHODS IF M=1*
     1,15(*.*),*M=*,I5,/)
12    FORMAT (1HO,*STARTING VALUE FOR VECTOR X(I)*,29(*.*),*XSTRT( 1)=*,
     1E16.8)
13    FORMAT (1HO,59X,*XSTRT(*,I2,*)=*,E16.8)
14    FORMAT (1HO,/,1HC,*TEST QUANTITIES TO BE USED IN NEW FLETCHER METH
     1OD*,12(*.*),*EPS( 1)=*,E16.8)
15    FORMAT (1HO,61X,*EPS(*,I2,*)=*,E16.8)
16    FORMAT (1HC,/,1HC,*TEST QUANTITY TO BE USED IN FLETCHER-POWELL MET
     1HOD*,14(*.*),*EPS1=*,E16.8)
17    FORMAT (1HO,/,1HC,*ESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIM
     1IZED*,14(*.*),*EST=*,E16.8)
18    FORMAT (1HO,*NONE OF THE OPTIMIZATION METHODS HAVE BEEN CALLED*,/,
     11X,*PLEASE CHECK THE VALUE OF MET*,/,1X,*REMAINDER*,/,1X,*MET=1
     2 NEW FLETCHER METHOD WOULD BE CALLED*,/,1X,*MET=2    FLETCHER-POWE
     3LL METHOD WOULD BE CALLED*)
      END
```

```
      SUBROUTINE FINAL (X,F,N)
C
CC        PRINTS THE RESULTS
CC        FOR THE OPTIMIZATION PROCESS
C
      DIMENSION X(1)
      COMMON /BLK/ KO
      WRITE (6,3)
      IF (KO.EQ.0) GO TO 1
      WRITE (6,4)
      GO TO 2
1     WRITE (6,5)
2     CONTINUE
      WRITE (6,6) F
      WRITE (6,7) (I,X(I),I=1,N)
      RETURN
C
3     FORMAT (1H1)
4     FORMAT (1HO,43X,*FOLLOWING IS THE OPTIMUM SOLUTION*,/,41X,*-------
     1--------------------------*)
5     FORMAT (1HO,44X,*RESULTS AT LAST ITERATION*/,45X,*----------------
     1---------*)
6     FORMAT (1HO,//,48X,*F =*,E16.8,/)
7     FORMAT (1HO,44X,*X(*,I2,*)=*,E16.8)
      END
```

```
      SUBROUTINE WRITE1 (N)
C
CC        PRINTS THE INTERMEDIATE RESULTS
CC
```

```
      WRITE (6,4)                                                       K    5
      GO TO (1,2), N                                                    K    6
    1 WRITE (6,5)                                                       K    7
      GO TO 3                                                           K    8
    2 WRITE (6,6)                                                       K    9
    3 CONTINUE                                                          K   10
      WRITE (6,7)                                                       K   11
      RETURN                                                            K   12
C                                                                       K   13
    4 FORMAT (1H1)                                                      K   14
    5 FORMAT (1HO,*OPTIMIZATION BY NEW FLETCHER METHOD*,/,1HO,*---------K   15
   1----------*)                                                        K   16
    6 FORMAT (1HO,*OPTIMIZATION BY FLETCHER-POWELL METHOD*,/,1HO,*------K   17
   1----------*)                                                        K   18
    7 FORMAT (1HO,*ITERATION*,2X,*FUNCTION*,6X,*TIME ELAPSED*,8X,*OBJECT K   19
   1IVE *,13X,*VARIABLE VECTOR X(I)*,9X,*GRADIENT VECTOR G(I)*,/1HO,*N  K   20
   2UMBER*,5X,*EVALUATIONS*,3X,*(SECONDS)*,11X,*FUNCTION*,/)            K   21
      END                                                               K   22-

      .....................................................................


      SUBROUTINE WRITE2 (X,N,G,F,NUMF,ITER,TIME)                        L    1
C                                                                       L    2
C         PRINTS THE INTERMEDIATE RESULTS                               L    3
C                                                                       L    4
      DIMENSION X(1), G(1)                                              L    5
      WRITE (6,1) ITER,NUMF,TIME,F,((X(I),G(I)),I=1,N)                  L    6
      RETURN                                                            L    7
C                                                                       L    8
    1 FORMAT (1HO,I5,7X,I5,5X,E16.8,3X,E16.8,12X,95(E16.8,13X,E16.8,/,70 L    9
   1X))                                                                 L   10
      END                                                               L   11-

HSBZOED  //// END OF LIST ////
```

APPENDIX IV

The Chebyshev polynomials

The Chebyshev polynomial of nth order, T_n, is defined

$$T_n(t) = \cos(n \arccos t), \quad -1 \leq t \leq 1$$

T_n has n zeros

$$t_k = \cos\ (2k-1)\ \frac{\pi}{2n}\ ,$$

$$= 1-2\ \sin^2(k-0.5)\ \frac{\pi}{2n}\ , \qquad k=1,2,..,n,$$

and n+1 extrema

$$t_k = \cos\ \frac{k\pi}{n}$$

$$= 1-2\ \sin^2\ \frac{k\pi}{2n}\ , \qquad k=0,1,...,n$$

over the interval [-1,1].

T_n may be transformed on any closed interval [a,b] by introducing the change in variable

$$x = \frac{b-a}{2}\ t + \frac{b+a}{2}$$

where $x \varepsilon [a,b]$.

The zeros of T_n in the transformed interval are in

$$x_k = a + (b-a)\sin^2(k-0.5)\ \frac{\pi}{2n}\ , \qquad k=1,2,...,n,$$

and extrema in

$$x_k = a + (b-a)\sin^2\ \frac{k\pi}{2n}\ , \qquad k=0,1,...,n.$$

The initial approximation for the Test 1 and Test 2 given in the Chapter V is obtained by rational interpolation where the zeros of the (k+1)st Chebyshev polynomial are used as supporting points. As a first trial for the points of the independent parameter the extrema of T_{n+1} were taken. The listing of a FORTRAN program which calculates the initial approximation and the starting point set is presented below.

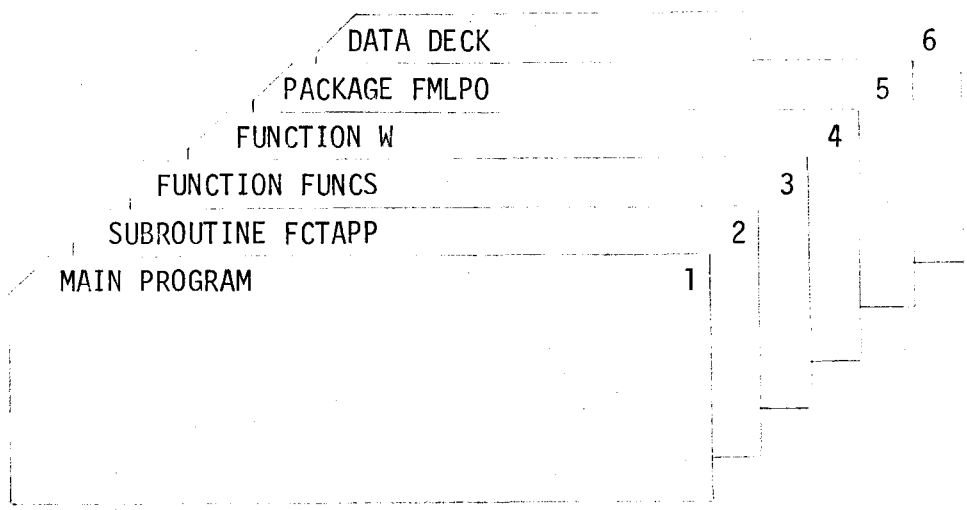
```
      DIMENSION A(10),G(10),Y(10),ASTRT(10),DET(100),X(50)
C
  100 FORMAT(8I10)
  200 FORMAT(////15X,*MATRIX IS SINGULAR IN SUBROUTINE SIMQ*//)
  300 FORMAT(8F10.5)
      READ(5,100)N,L,M
      READ(5,300)XA,XB
      PI=3.14159
      XAB=XB-XA
      X(2)=XA
      II=1
      J=N+1
C
C         COMPUTATION OF AN INITIAL GUESS X(I) OF THE INDEPENDENT
C         VARIABLE IN THE EXTREME POINTS OF THE CHEBYSHEV POLYNOMIAL
C         T(N+1 )
C
      CONST=PI/(2.*J)
      DO 10 I=1,J
      Q=SIN(I*CONST)
      X(I+2)=Q*Q*XAB+XA
   10 CONTINUE
      X(N+3)=XB
C
C         CALCULATION OF THE STARTING VALUES OF THE COEFFICIENTS OF
C         THE RATIONAL FUNCTION BY RATIONAL INTERPOLATION IN THE
C         ZEROS OF THE CHEBYSHEV POLINOMIAL T(K+1)
C
      K=L+M+1
   15 CONST=PI/(2.*K)
      DO 20 I=1,K
      Q=SIN((I-0.5)*CONST)
      Y(I)=Q*Q*XAB+XA
      G(I)=FCT(Y(I))
   20 CONTINUE
      DO 130 I=1,K
      DET(I)=1.
      II=I
      IF(L)115,115,105
  105 DO 110 J=1,L
      IA=II
      II=II+K
  110 DET(II)=DET(IA)*Y(I)
  115 II=II+K
      IF(M-1)130,117,117
  117 DET(II)=-G(I)*Y(I)
      IF(M-2)130,118,118
  118 DO 120 J=2,M
      IB=II
      II=II+K
  120 DET(II)=DET(IB)*Y(I)
  130 CONTINUE
      CALL SIMQ(DET,G,K,KS)
      IF(KS)134,134,150
  134 L1=L+1
      DO 135 I=1,L1
  135 A(I)=G(I)
      IF(M)137,137,138
  138 DO 136 J=1,M
      I=L1+J
      A(I)=G(I)
  136 CONTINUE
  137 CONTINUE
      DO 30 I=1,K
      ASTRT(I)=G(I)
   30 CONTINUE
      GO TO 151
  150 WRITE(6,200)
  151 CONTINUE
      CALL EXIT
```

APPENDIX V

Usage of FMLPO (Function Minimization for Least pth Objectives)

Purpose To minimize the objective function of k variables a_i defined as the generalized discrete least pth objective (6.13), using gradient methods.

How to use Set the input deck as follows:



1. Main program

Write the main program as indicated below:

(i) Dimension the following arrays

A(K), ASTRT(K), G(K), Y(K), PY(K), DUM1(K), DUM2(K), GRAD(K), EPS(K), H(M), XX(3, NINT), NUMB(N), INUMB(N), X(N), X1(N), ERROR(N), EHELP(N), AP(N), IPA(ITER)

where

K is the number of variable parameters,

M = K·(K+7)/2,

NINT is the number of intervals,

N is the total discrete point set of independent parameter , from all intervals, and

ITER is the maximum number of allowable usage of the optimization method.

(ii) Call the subroutine FMLPO as follows:

CALL FMLPO (A, ASTRT, G, Y, PY, DUM1, DUM2, EPS, H, GRAD, NUMB, XX, X, X1, ERROR, EHELP, AP, INUMB, IPA)

(iii) Add STOP and END cards.

2. Subroutine FCTAPP

Subroutine which defines the approximating function in each interval and calculates its gradients with respect to variable vector $\underset{\sim}{a}$.

Write subroutine FCTAPP as follows:

(i) SUBROUTINE FCTAPP (X, K, A, APP, GRAD, IINT, INDIC)

DIMENSION A(1), GRAD(1).

where X, K, A, IINT and INDIC are input, and APP and GRAD are output variables.

INDIC may have a value 1 or 2 and indicates whether the approximating function in the IINT interval APP=$F(\underset{\sim}{A}, X)$, or all its gradients $GRAD(I)=\frac{\partial F(\underset{\sim}{A}, X)}{\partial A(I)}$, I=1,K should be calculated, respectively.

(ii) Write the approximating function APP and all its gradients GRAD(I), I = 1, K for each interval IINT = 1, NINT.

(iii) Add RETURN and END cards.

3. Function FUNCS

Function subprogram FUNCS defines upper and lower specified function $S_u(x)$ and $S_\ell(x)$, respectively, in various intervals.

```
      FUNCTION FUNCS (X, IINT)
      GO TO (1,2,...,NINT), IINT
1     FUNCS = S_u/ℓ (x)
      RETURN
2
.
.
.
NINT  FUNCS = S_u/ℓ (x)
      RETURN
      END
```

where IINT = 1, NINT is a current interval.

Note If the upper and lower specified functions are defined for the same set of the independent variable x, consider the common interval (or subinterval) twice.

4. Function W

Function subprogram W defines an upper and lower positive weighting function $W_u(x)$ and $W_\ell(x)$, respectively, in various intervals.

Write subprogram W as follows:

```
        FUNCTION W(X,IINT)
        GO TO (1,2,...,NINT), IINT
1       W = w_{u/ℓ}(x)
        RETURN
2
.
.
.
NINT    W = w_{u/ℓ}(x)
        RETURN
        END
```

where IINT = 1,2,...,NINT is an interval under consideration.

An example which shows how to write these subroutines is the same as in Test 2 in Chapter VI, but with constraints on parameter a_2 such that

$$0 \leq a_2 \leq 1.$$

The listings which the user has to write follow.

```
      SUBROUTINE FCTAPP(X,K,A,APP,GRAD,IINT,INDIC)
C
C           SUBROUTINE WHICH CALCULATES APPROXIMATING
C          FUNCTION AND ITS GRADIENTS WITH RESPECT TO
C          VARIABLE PARAMETERS
C
      DIMENSION A(1),GRAD(1)
      GO TO(100,200),INDIC
  100 GO TO (1,1,2,2),IINT
    1 APP=A(3)/A(2)*EXP(-A(1)*X)*SIN(A(2)*X)
      RETURN
    2 APP=A(2)
      RETURN
  200 GO TO (3,3,4,4),IINT
    3 HP1=1./A(2)*EXP(-A(1)*X)
      HP2=HP1*SIN(A(2)*X)
      HP3=HP1*COS(A(2)*X)
      GRAD(1)=-APP*X
      GRAD(2)=-1./A(2)*APP+A(3)*X*HP3
      GRAD(3)=HP2
      RETURN
    4 GRAD(1)=0.
      GRAD(2)=1.
      GRAD(3)=0.
      RETURN
      END
```

.....

```
      FUNCTION FUNCS(X,IINT)
C
C          FUNCTION SUBPROGRAM WHICH DEFINES
C          UPPER AND LOWER
C          SPECIFIED FUNCTION
C
      GO TO(1,1,2,3),IINT
    1 FUNCS=3./20.*EXP(-X)+1./52.*EXP(-5.*X)-EXP(-2.*X)/65.*(3.*SIN
     *(2.*X)+11*COS(2.*X))
      RETURN
    2 FUNCS=1.0
      RETURN
    3 FUNCS=0.0
      RETURN
      END
```

```
      ..........................................

      FUNCTION W(X,IINT)
C
C          FUNCTION SUBPROGRAM WHICH DEFINES
C          UPPER AND LOWER
C          WEIGHTING FUNCTION
C
      GO TO (1,1,1,1),IINT
    1 W=1.
      RETURN
      END
```

6. Data deck

Parameters to be supplied as data are defined below:

K	The number of independent variable parameters $\underset{\sim}{a}$.
NINT	The total number of upper and lower intervals.
NUMB(I), I=1, NINT	Number of subintervals in each interval of independent parameters.
XX(1,I), I=1, NINT	The left end point of ith interval.
XX(2,I), I=1, NINT	The right end point of ith interval.
XX(3,I), I=1, NINT	Numbers in floating point which supply informations on the specified function in the ith interval: set XX(3,I)=1. for the upper specification and set XX(3,I)=-1. for the lower specification.
IREAD	Integer which denotes whether or not the discrete set of points in each interval will be read. If IREAD=0 the discrete set of points will be set equidistantly in each interval with NUMB(I) sub-intervals in the ith interval. If IREAD=1 the discrete point set will be read.

$X(I),\ I=1,\sum_{I=1}^{NINT}(NUMB(I)+1)$	Discrete point set of the independent parameter.
KSI	The artificial margin ξ.
ASTRT(I), I=1, K	Starting values for a vector containing the K variable parameters.
IGRDCH	Gradients to be checked if IGRDCH=1; it should be set to 0 if gradients are not to be checked.
MET	Optimization method to be called: if MET=1 Fletcher method will be called; if MET=2 Fletcher-Powell method will be called.
MAX	Maximum number of permissible iterations.
ITER	Has already been defined in the main program as a length of the working array.
IPA(I), I=1, ITER	Vector containing the values of p for different least pth objective.
IOPT	Denotes how many times the optimization is repeated with different starting points and/or different optimization techniques.
IPRINT	Intermediate output is printed out every IPRINT iterations; it should be set to 0 if no intermediate output is desired.

IDATA	Input data is printed out if IDATA=1; it should be set to 0 if input data is not to be printed out.
EST	Minimum estimated value of the objective function.
EPS(I), I=1, K	Small test quantities used by the Fletcher method.
EPS1	Small test quantity used by the Fletcher-Powell method.
DIF	Small test quantity used by the subroutine FMLPO.

Setting up the data deck

	Conditions	Number of cards	Parameters	Type	Format
	-	1	K,NINT,IOPT,ITER,IREAD. IGRDCH	INTEGER	6I10
	-	I=1, NINT	NUMB(I), (XX(J,I), J=1,3)	1 INTEGER, 3 REALS	I10, 3E16.8
	IREAD=1	As many as required by NUMB(I), I=1, NINT	X(J), J=1 (NUMB(I)+1)	REAL	5E16.8
	-	1	EST, DIF, KSI	REAL	3E16.8
↑	-	As many as required by K	ASTRT(I), I=1,K	REAL	5E16.8
IOPT times	-	1	MET, MAX, IPRINT, IDATA	INTEGER	4I10
	MET=1	As many as required by K	EPS(I), I=1,K	REAL	5E16.8
	MET=2	1	EPSI	REAL	5E16.8
	-	As many as required by ITER	IPA(I), I=1, ITER	INTEGER	8I10
↓					

Recommended values for some of the parameters

MAX = 100;

EPS(I), I=1,K, each 10^{-6},

DIF = 10^{-4}

EST A lower bound of the minimum value of the objective function may be obtained from physical reasons. If the true minimum is not known, for the case when the specification is violated EST=0 is convenient, and when the specification is satisfied choose EST sufficiently negative.

For the example for which listings were shown the data deck is presented below. The Fletcher-Powell optimization method is called.

```
  3          4          1          5          0          0
 50       0.E 00     10.E 00     1.E 00
 50       0.E 00     10.E 00    -1.E 00
  0       2.0E 1      2.0E 1     1.0E 00
  0       3.0E 1      3.0E 1    -1.0E 00
-1.0E 00  1.0E -4    20.0E -3
 1.  E00   1.  E00    1.  E00
  2   100   1    1
 1.0E -6
  2    10   100   1000   10000
```

APPENDIX VI

Listing of the Package FMLPO

```
      SUBROUTINE FMLPO (A,XSTRT,G,Y,PY,DUM1,DUM2,EPS,H,GRAD,NUMB,XX,X,X1   A    1
     1,ERROR,EHELP,AP,INUMB,IPA)                                           A    2
C                                                                          A    3
CCC       SUBROUTINE WHICH COORDINATES THE OTHER                           A    4
CCC       SUBROUTINES IN THE PACKAGE FMLPO                                 A    5
C                                                                          A    6
      EXTERNAL FUNCS,W,FCT,FUNGT                                           A    7
      LOGICAL CONV,UNITH                                                   A    8
      DIMENSION A(1), G(1), Y(1), PY(1), XSTRT(1), DUM1(1), DUM2(1), EPS   A    9
     1(1), H(1), GRAD(1), NUMB(1), XX(3,1), X(1), X1(1), ERROR(1), EHELP   A   10
     2(1), AP(1), INUMB(1), IPA(1)                                         A   11
      COMMON /BLK/ KO                                                      A   12
      EPR(Z)=EPSNP(Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,1)                    A   13
      UNITH=.TRUE.                                                         A   14
      READ (5,36) N1,NINT,IOPT,ITER,IREAD,IGROCH                           A   15
      DO 1 I=1,NINT                                                        A   16
      READ (5,38) NUMB(I),(XX(J,I),J=1,3)                                  A   17
1     CONTINUE                                                             A   18
      K=0                                                                  A   19
      IF (IREAD.EQ.0) IREAD=2                                              A   20
      GO TO (2,4),IREAD                                                    A   21
2     DO 3 J=1,NINT                                                        A   22
      K=K+1                                                                A   23
      KL=K+NUMB(J)                                                         A   24
      READ (5,40) (X(I),I=K,KL)                                            A   25
3     K=KL                                                                 A   26
4     READ (5,40) EST,DIF,PSI                                              A   27
      WRITE (6,37) PSI                                                     A   28
      WRITE (6,44)                                                         A   29
      READ (5,40) (XSTRT(I),I=1,N1)                                        A   30
      DO 5 I=1,N1                                                          A   31
5     A(I)=XSTRT(I)                                                        A   32
      K=0                                                                  A   33
      GO TO (6,9),IREAD                                                    A   34
6     DO 8 J=1,NINT                                                        A   35
      IINT=J                                                               A   36
      WRITE (6,49) IINT                                                    A   37
      K=K+1                                                                A   38
      KL=K+NUMB(J)                                                         A   39
      DO 7 I=K,KL                                                          A   40
      EPROR(I)=ERR(X(I))                                                   A   41
      L=I-K+1                                                              A   42
      WRITE (6,48) L,X(I),ERROR(I)                                         A   43
      EHELP(I)=ERROR(I)*XX(3,J)                                            A   44
      AP(I)=APP                                                            A   45
7     CONTINUE                                                             A   46
      K=KL                                                                 A   47
8     CONTINUE                                                             A   48
      GO TO 11                                                             A   49
9     DO 10 J=1,NINT                                                       A   50
      IINT=J                                                               A   51
      WRITE (6,49) IINT                                                    A   52
      L=NUMB(J)+1                                                          A   53
      IF (NUMB(J).EQ.0) Z=XX(1,J)                                          A   54
      DO 10 I=1,L                                                          A   55
      IF (NUMB(J).GT.0) Z=XX(1,J)+(XX(2,J)-XX(1,J))*(I-1)/NUMB(J)          A   56
      ER=ERR(Z)                                                            A   57
      WRITE (6,48) I,Z,ER                                                  A   58
      K=K+1                                                                A   59
      ERROR(K)=ERR(Z)                                                      A   60
      EHELP(K)=ERROR(K)*XX(3,J)                                            A   61
      X(K)=Z                                                               A   62
10    AP(K)=APP                                                            A   63
11    EMAX=EHELP(1)                                                        A   64
      DO 12 M=2,K                                                          A   65
      EMAX=AMAX1(EMAX,EHELP(M))                                            A   66
12    CONTINUE                                                             A   67
      WRITE (6,46)                                                         A   68
      WRITE (6,47)                                                         A   69
      WRITE (6,45) EMAX                                                    A   70
      CALL ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,2,NUMB,XX,X,X1,EPROR,EHELP,AP   A   71
     1,EMAX,N,INUMB,NINT,IP)                                               A   72
      WRITE (6,42)                                                         A   73
      WRITE (6,43)                                                         A   74
      WRITE (6,48) (J,X1(J),ERROR(J),J=1,N)                                A   75
C                                                                          A   76
CC        DATA FOR THE OPTIMALITY                                          A   77
CC        FOR THE OPTIMIZATION METHOD USED                                 A   78
C                                                                          A   79
      DO 35 K=1,IOPT                                                       A   80
      KR=1                                                                 A   81
      IF (K-1) 14,14,13                                                    A   82
13    READ (5,40) (XSTRT(I),I=1,N1)                                        A   83
14    READ (5,36) MET,MAX,IPRINT,IDATA                                     A   84
      IF (MET.EQ.1) READ (5,40) (EPS(I),I=1,N1)                            A   85
      IF (MET.EQ.2) READ (5,40) EPS1                                       A   86
      READ (5,36) (IPA(I),I=1,ITER)                                        A   87
      DO 34 KK=1,ITER                                                      A   88
C                                                                          A   89
CC        OPTIMIZATION                                                     A   90
C                                                                          A   91
      IP=IPA(KK)                                                           A   92
      IF (KK.GT.1) FF=F                                                    A   93
      IF (KR.EQ.0) GO TO 15                                                A   94
15    DO 16 I=1,N1                                                         A   95
      A(I)=XSTRT(I)                                                        A   96
```

```
16    CONTINUE                                                          A   97
      IF (IGROCH.NE.1) GO TO 17                                         A   98
      CALL GROGHK (N1,A,G,PY,Y,GRAD,APP,PSI,NUMB,XX,X,X1,ERROR,EHELP,AP,A   99
     1EMAX,N,INUMB,NINT,IP,DUM1)                                        A  100
17    IF (KR.EQ.0) GO TO 18                                             A  101
      IF (IDATA.EQ.0) GO TO 18                                          A  102
      CALL INPUT (MET,M,MAX,N1,IPRINT,IDATA,EPS1,EST,EPS,XSTRT)         A  103
18    IF (MET.EQ.0) MET=4                                               A  104
      INDEX=0                                                           A  105
      GO TO (19,24,30,29), MET                                          A  106
19    IF (IPRINT.EQ.0) GO TO 20                                         A  107
      CALL WRITE1 (1)                                                   A  108
20    CALL SECOND (T1)                                                  A  109
      IF (KR.NE.0) GO TO 22                                             A  110
      DO 21 I=1,N1                                                      A  111
      A(I)=DUM1(I)                                                      A  112
21    CONTINUE                                                          A  113
22    CALL FMNFG (N1,A,F,G,H,UNITH,EST,EPS,MAX,IPRINT,IEXIT,GRAD,APP,PSI A  114
     1,NUMB,XX,X,X1,ERROR,EHELP,AP,EMAX,N,INUMB,NINT,IP)                A  115
      DO 23 I=1,N1                                                      A  116
      DUM1(I)=A(I)                                                      A  117
23    CONTINUE                                                          A  118
      CALL SECOND (T2)                                                  A  119
      CALL FINAL (A,F,N1)                                               A  120
      T=T2-T1                                                           A  121
      WRITE (6,39) T                                                    A  122
      GO TO 29                                                          A  123
24    IF (IPRINT.EQ.0) GO TO 25                                         A  124
      CALL WRITE1 (2)                                                   A  125
25    CALL SECOND (T1)                                                  A  126
      IF (KR.NE.0) GO TO 27                                             A  127
      DO 26 I=1,N1                                                      A  128
      A(I)=DUM2(I)                                                      A  129
26    CONTINUE                                                          A  130
27    CALL FMFPG (FUNGT,N1,A,F,G,EST,EPS1,MAX,IER,H,IPRINT,GRAD,APP,PSI, A  131
     1NUMB,XX,X,X1,ERROR,EHELP,AP,EMAX,N,INUMB,NINT,IP)                 A  132
      DO 28 I=1,N1                                                      A  133
      DUM2(I)=A(I)                                                      A  134
28    CONTINUE                                                          A  135
      CALL SECOND (T2)                                                  A  136
      CALL FINAL (A,F,N1)                                               A  137
      T=T2-T1                                                           A  138
      WRITE (6,39) T                                                    A  139
29    INDEX=INDEX+1                                                     A  140
C                                                                       A  141
30    KP=0                                                              A  142
      WRITE (6,41) IP                                                   A  143
      WRITE (6,44)                                                      A  144
      KN=0                                                              A  145
      KQ=0                                                              A  146
      DO 32 J=1,NINT                                                    A  147
      IINT=J                                                            A  148
      WRITE (6,49) IINT                                                 A  149
      KQ=KQ+1                                                           A  150
      KL=KQ+NUMB(J)                                                     A  151
      DO 31 I=KQ,KL                                                     A  152
      L=I-KQ+1                                                          A  153
      ER=ERROR(X(I))                                                    A  154
      WRITE (6,48) L,X(I),ER                                            A  155
      KN=KN+1                                                           A  156
31    CONTINUE                                                          A  157
      KQ=KL                                                             A  158
32    CONTINUE                                                          A  159
      WRITE (6,42)                                                      A  160
      WRITE (6,43)                                                      A  161
      WRITE (6,48) (J,X1(J),ERROR(J),J=1,N)                             A  162
      WRITE (6,46)                                                      A  163
      WRITE (6,47)                                                      A  164
      WRITE (6,45) EMAX                                                 A  165
      IGROCH=IGROCH+2                                                   A  166
      IF (KK-1) 34,34,33                                                A  167
33    FTST=ABS((FF-F)/FF)                                               A  168
      IF (FTST.LT.DIF) GO TO 35                                         A  169
34    CONTINUE                                                          A  170
35    CONTINUE                                                          A  171
      RETURN                                                            A  172
C                                                                       A  173
36    FORMAT (8I10)                                                     A  174
37    FORMAT (1H1,20X,*KSI=*,E23.12/////)                               A  175
38    FORMAT (I10,3D16.8)                                               A  176
39    FORMAT (1HO,//25X,*EXECUTION TIME IN SECONDS=*,F10.5)             A  177
40    FORMAT (5E15.8)                                                   A  178
41    FORMAT (1H1,19X,*P  =*,I7/////)                                   A  179
42    FORMAT (//41X,*ERROR USED IN*)                                    A  180
43    FORMAT (40X,*OBJECTIVE FUNCTION*/)                                A  181
44    FORMAT (9X,*N*,6X,*INDEPENDENT VARIABLE*,8X,*ERRORS*)             A  182
45    FORMAT (13),E23.12)                                               A  183
46    FORMAT (//20X,*VALUE OF*)                                         A  184
47    FORMAT (18X,*MAXIMUM ERROR*/)                                     A  185
48    FORMAT (I9,3X,2E23.12)                                            A  186
49    FORMAT (/20X,*INTERVAL *,I2/)                                     A  187
      END                                                               A  188-
```

.....

```
      FUNCTION FCT (Z,FUNCS,W,IINT,PSI,XX)                                  B    1
C                                                                           B    2
CC        FUNCTION SUBPROGRAM WHICH DEFINES                                 B    3
CC        MODIFIED UPPER AND LOWER                                          B    4
CC        SPECIFIED FUNCTION                                                B    5
C                                                                           B    6
      EXTERNAL FUNCS,W                                                      B    7
      DIMENSION XX(3,1)                                                     B    8
      FCT=FUNCS(Z,IINT)+PSI*XX(3,IINT)/W(Z,IINT)                            B    9
      RETURN                                                                B   10
      END                                                                   B   11-

     ..................................................................

      FUNCTION EPSNP (Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,IPOINT)             C    1
C                                                                           C    2
CC        FUNCTION SUBPROGRAM WHICH CALCULATES                              C    3
CC        UPPER AND LOWER WEIGHTED ERROR FUNCTION                           C    4
C                                                                           C    5
      EXTERNAL FUNCS,W,FCT                                                  C    6
      DIMENSION A(1), GRAD(1), XX(3,1)                                      C    7
      IF (IPOINT) 1,2,1                                                     C    8
1     CONTINUE                                                              C    9
      CALL FCTAPP (Z,N1,A,APP,GRAD,IINT,1)                                  C   10
2     CONTINUE                                                              C   11
      IF (PSI) 3,4,3                                                        C   12
3     EPSNP=(APP-FCT(Z,FUNCS,W,IINT,PSI,XX))*W(Z,IINT)                      C   13
      RETURN                                                                C   14
4     EPSNP=(APP-FUNCS(Z,IINT))*W(Z,IINT)                                   C   15
      RETURN                                                                C   16
      END                                                                   C   17-

     ..................................................................

      SUBROUTINE ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,INDIC,NUMB,XX,X,X1,ERRO    D    1
     1R,EHELP,AP,EMAX,N,INUMB,NINT,IP)                                      D    2
C                                                                           D    3
CC        SUBROUTINE WHICH SELECTS THE WEIGHTED                             D    4
CC        ERROR FUNCTION OF INTEREST FOR                                    D    5
C         THE OBJECTIVE FUNCTION                                            D    6
C                                                                           D    7
      EXTERNAL FUNCS,W,FCT                                                  D    8
      DIMENSION A(1), GRAD(1), NUMB(1), XX(3,1), X(1), X1(1), EPROR(1),     D    9
     1EHELP(1), AP(1), INUMB(1)                                             D   10
      EPR(Z)=EPSNP(Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,IPOINT)                D   11
      GO TO (1,9), INDIC                                                    D   12
1     CONTINUE                                                              D   13
      IPOINT=1                                                              D   14
      K=0                                                                   D   15
      KL=0                                                                  D   16
      DO 7 J=1,NINT                                                         D   17
      IINT=J                                                                D   18
      IF (J.EQ.1) GO TO 2                                                   D   19
      KL=KL+L                                                               D   20
2     L=NUMB(J)+1                                                           D   21
      DO 6 I=1,L                                                            D   22
      K=K+1                                                                 D   23
      IF (J.EQ.1) GO TO 5                                                   D   24
      DO 4 KK=1,KL                                                          D   25
      IF (X(K)-X(KK)) 4,3,4                                                 D   26
3     AP(K)=AP(KK)                                                          D   27
      APP=AP(K)                                                             D   28
      IPOINT=0                                                              D   29
      GO TO 5                                                               D   30
4     CONTINUE                                                              D   31
5     EPROR(K)=EPR(X(K))                                                    D   32
      EHELP(K)=EPROR(K)*XX(3,J)                                             D   33
      IF (IPOINT.NE.0) AP(K)=APP                                            D   34
      IPOINT=1                                                              D   35
6     CONTINUE                                                              D   36
7     CONTINUE                                                              D   37
      EMAX=EHELP(1)                                                         D   38
      DO 8 M=2,K                                                            D   39
      EMAX=AMAX1(EMAX,EHELP(M))                                             D   40
8     CONTINUE                                                              D   41
9     CONTINUE                                                              D   42
      IF (EMAX) 10,11,11                                                    D   43
10    IP=-IABS(IP)                                                          D   44
      GO TO 12                                                              D   45
11    IP=IABS(IP)                                                           D   46
12    K=0                                                                   D   47
      N=0                                                                   D   48
      INUMB(1)=0                                                            D   49
      DO 16 J=1,NINT                                                        D   50
      IINT=J                                                                D   51
      L=NUMB(J)+1                                                           D   52
      DO 15 I=1,L                                                           D   53
      K=K+1                                                                 D   54
```

```
      IF (IP) 14,13,13                                                  D   55
13    IF (EHELP(K)) 15,14,14                                            D   56
14    N=N+1                                                             D   57
      X1(N)=X(K)                                                        D   58
      ERROR(N)=ERROP(K)                                                 D   59
      EHELP(N)=AP(K)                                                    D   60
15    CONTINUE                                                          D   61
      INUMB(J+1)=N                                                      D   62
16    CONTINUE                                                          D   63
      RETURN                                                            D   64
      END                                                               D   65-

      ..................................................................

      SUBROUTINE FUNGT (N1,A,OBJ,G,GRAD,APP,PSI,NUMB,XX,X,X1,ERROR,EHELP  E    1
     1,AP,EMAX,N,INUMB,NINT,IP)                                         E    2
C                                                                        E    3
CCC       SUBROUTINE WHICH COMPUTES THE OBJECTIVE FUNCTION               E    4
CCC       AND ITS GRADIENTS W.R.T. THE VARIABLE PARAMETERS               E    5
C         IN THE LEAST P-TH SENSE                                        E    6
C                                                                        E    7
      EXTERNAL FUNCS,W,FCT                                               E    8
      DIMENSION A(1), GRAD(1), NUMB(1), XX(3,1), X(1), X1(1), ERROR(1),  E    9
     1EHELP(1), AP(1), INUMB(1), G(1)                                   E   10
      OBJP=0.                                                           E   11
      GRADP=0.                                                          E   12
      DO 1 K=1,N1                                                       E   13
      G(K)=0.                                                           E   14
1     CONTINUE                                                          E   15
      CALL ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,1,NUMB,XX,X,X1,ERROR,EHELP,AP E   16
     1,EMAX,N,INUMB,NINT,IP)                                            E   17
      DO 7 I=1,N                                                        E   18
      Z=X1(I)                                                           E   19
      DEL=ERROR(I)/EMAX                                                 E   20
      OBJI=DEL**IP                                                      E   21
      GRADI=DEL**(IP-1)                                                 E   22
      OBJP=OBJP+OBJI                                                    E   23
      DO 4 J=1,NINT                                                     E   24
      IF (I-INUMB(J+1)) 2,2,4                                           E   25
2     IF (I-INUMB(J)) 4,4,3                                             E   26
3     IINT=J                                                            E   27
      GO TO 5                                                           E   28
4     CONTINUE                                                          E   29
5     CONTINUE                                                          E   30
      APP=EHELP(I)                                                      E   31
      CALL FCTAPP (Z,N1,A,APP,GRAD,IINT,2)                              E   32
      DO 6 K=1,N1                                                       E   33
      GRAD(K)=GRADI*W(Z,IINT)*GRAD(K)                                   E   34
      G(K)=G(K)+GRAD(K)                                                 E   35
6     CONTINUE                                                          E   36
7     CONTINUE                                                          E   37
      PR=1./IP                                                          E   38
      OBJ=EMAX*(OBJP**PR)                                               E   39
      GRP=OBJP**(PR-1.)                                                 E   40
      DO 8 K=1,N1                                                       E   41
      G(K)=GRP*G(K)                                                     E   42
8     CONTINUE                                                          E   43
      RETURN                                                            E   44
      END                                                               E   45-

      ..................................................................

      SUBROUTINE GRDCHK (N,X,G,PY,Y,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHE  F    1
     1LP,AP,EMAX,NP,INUMB,NINT,IP,DUM1)                                 F    2
C                                                                        F    3
CCC       SUBROUTINE WHICH CHECKS THE GRADIENTS                          F    4
CCC       W.R.T. ALL VARIABLE PARAMETERS                                 F    5
C                                                                        F    6
      DIMENSION X(1), G(1), PY(1), Y(1), GRAD(1), NUMB(1), XX(3,1), XP(1 F    7
     1), X1(1), ERROR(1), EHELP(1), AP(1), INUMB(1), DUM1(1)            F    8
      CALL FUNGT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX F    9
     1,NP,INUMB,NINT,IP)                                                F   10
      DO 1 I=1,N                                                        F   11
      DELX=1.E-4*X(I)                                                   F   12
      X(I)=X(I)+DELX                                                    F   13
      CALL FUNGT (N,X,FNEW,PY,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP, F   14
     1EMAX,NP,INUMB,NINT,IP)                                            F   15
      Y(I)=(FNEW-F)/DELX                                                F   16
      DUM1(I)=Y(I)                                                      F   17
      X(I)=X(I)-DELX                                                    F   18
1     CONTINUE                                                          F   19
      DO 2 I=1,N                                                        F   20
      IF (ABS(Y(I)).LT.1.E-20) DUM1(I)=1.E-20                           F   21
      PY(I)=ABS((Y(I)-G(I))/DUM1(I))*100.                               F   22
2     CONTINUE                                                          F   23
      WRITE (6,6)                                                       F   24
      WRITE (6,7)                                                       F   25
      WRITE (6,8) (I,X(I),I=1,N)                                        F   26
      WRITE (6,9)                                                       F   27
      DO 3 I=1,N                                                        F   28
```

```
      WRITE (6,10) G(I),Y(I),PY(I)                                      F   29
3     CONTINUE                                                          F   30
      DO 4 I=1,N                                                        F   31
      IF (PY(I).GT.1.) GO TO 5                                          F   32
4     CONTINUE                                                          F   33
      WRITE (6,11)                                                      F   34
      RETURN                                                            F   35
5     WRITE (6,12)                                                      F   36
      CALL EXIT                                                         F   37
C                                                                       F   38
6     FORMAT (1H1)                                                      F   39
7     FORMAT (1HO,5X,*GRADIENTS CHECKING*,/,6X,18(*-*),//,6X,*GRADIENTS F   40
     1HAVE BEEN CHECKED AT THE FOLLOWING POINT*/)                       F   41
8     FORMAT (10X,*X(*,I2,*)=*,E16.8)                                   F   42
9     FORMAT (///,1HO,5X,*ANALYTICAL GRADIENTS*,5X,*NUMERICAL GRADIENTS* F   43
     1,5X,*PERCENTAGE ERROR*,/)                                         F   44
10    FORMAT (1HO,3X,3(E16.8,3X))                                       F   45
11    FORMAT (1HO,//,6X,*GRADIENTS ARE O. K.*)                          F   46
12    FORMAT (1HO,//,6X,*YOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRADIE F   47
     1NTS ARE INCORRECT*,/6X,*PLEASE CHECK IT AGAIN*)                   F   48
      END                                                               F   49-

      ..................................................................


      SUBROUTINE FMNFG (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT,GRAD  G    1
     1,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX,NP,INUMB,NINT,IP)        G    2
C                                                                       G    3
C         PURPOSE                                                       G    4
C            TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES G    5
C            ASSUMING THAT ITS GRADIENTS CAN BE CALCULATED EXPLICITLY   G    6
C            BY THE METHOD OF FLETCHER                                  G    7
C                                                                       G    8
C            THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE           G    9
C            R. FLETCHER, A NEW APPROACH TO VARIABLE METRIC ALGORITHMS, G   10
C            COMP. JOURNAL, VOL.13, 1970, PP.317-322.                   G   11
C                                                                       G   12
      DIMENSION X(1), G(1), H(1), EPS(1), GRAD(1), NUMB(1), XX(3,1), XP(  G   13
     11), X1(1), ERROR(1), EHELP(1), AP(1), INUMB(1)                    G   14
      LOGICAL CONV,UNITH                                                G   15
      COMMON /BLK/ KO                                                   G   16
      CALL SECOND (T3)                                                  G   17
      KO=0                                                              G   18
      CALL FUNGT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX  G   19
     1,NP,INUMB,NINT,IP)                                                G   20
      IF (F.LT.FEST) GO TO 23                                           G   21
      NFNS=1                                                            G   22
      ITN=0                                                             G   23
      STEP=1.                                                           G   24
      IDX=N                                                             G   25
      IDG=N+N                                                           G   26
      IH=IDG+N                                                          G   27
      IF (.NOT.UNITH) GO TO 2                                           G   28
      IJ=IH+1                                                           G   29
      DO 1 I=1,N                                                        G   30
      DO 1 J=I,N                                                        G   31
      H(IJ)=0.                                                          G   32
      IF (I.EQ.J) H(IJ)=1.0                                             G   33
1     IJ=IJ+1                                                           G   34
2     CONV=.TRUE.                                                       G   35
      GDX=0.                                                            G   36
      DO 6 I=1,N                                                        G   37
      Z=0.                                                              G   38
      IJ=IH+I                                                           G   39
      IF (I.EQ.1) GO TO 4                                               G   40
      II=I-1                                                            G   41
      DO 3 J=1,II                                                       G   42
      Z=Z-H(IJ)*G(J)                                                    G   43
      IJ=IJ+N-J                                                         G   44
3     CONTINUE                                                          G   45
4     DO 5 J=I,N                                                        G   46
      Z=Z-H(IJ)*G(J)                                                    G   47
      IJ=IJ+1                                                           G   48
5     CONTINUE                                                          G   49
      IF (ABS(Z).GT.EPS(I)) CONV=.FALSE.                                G   50
      H(IDX+I)=Z                                                        G   51
      GDX=GDX+G(I)*Z                                                    G   52
6     CONTINUE                                                          G   53
C                                                                       G   54
      IF (IPRINT.EQ.0) GO TO 7                                          G   55
      IF (MOD(ITN,IPRINT).NE.0) GO TO 7                                 G   56
      CALL SECOND (T4)                                                  G   57
      TIME=T4-T3                                                        G   58
      CALL WRITE2 (X,N,G,F,NFNS,ITN,TIME)                               G   59
7     IEXIT=1                                                           G   60
      IF (CONV) GO TO 24                                                G   61
      IEXIT=2                                                           G   62
      IF (GDX.GE.0.) GO TO 24                                           G   63
      Z=1.                                                              G   64
      IF (ITN.LT.N.AND.UNITH) Z=STEP                                    G   65
      W=2.*(FEST-F)/GDX                                                 G   66
      IF (W.LT.Z) Z=W                                                   G   67
      STEP=Z                                                            G   68
8     GDX=GDX*Z                                                         G   69
```

```
      DO 9 I=1,N                                                        GG  70
      H(IDX+I)=H(IDX+I)*Z                                               GG  71
      X(I)=X(I)+H(IDX+I)                                                GG  72
    9 CONTINUE                                                          GG  73
      CALL FUNCT (N,X,FP,H,GRAD,APP,PSI,NLMB,XX,XP,X1,ERROR,EHELP,AP,EMA GG  74
     1X,NB,INUMB,NIMT,IP)                                               GG  75
      IF (FP.LT.FEST) GO TO 23                                          GG  76
      NFNS=NFNS+1                                                       GG  77
      IEXIT=3                                                           GG  78
      IF (ITN.EQ.MAXFN) GO TO 24                                        GG  79
      GPDX=0.                                                           GG  80
      DO 10 I=1,N                                                       GG  81
      H(IDG+I)=H(I)-G(I)                                                GG  82
      GPDX=GPDX+H(I)*H(IDX+I)                                           GG  83
   10 CONTINUE                                                          GG  84
      DGDX=GPDX-GDX                                                     GG  85
      IF (F.GT.FP-.0001*GDX) GO TO 12                                   GG  86
      IEXIT=4                                                           GG  87
      IF (GPDX.LT.0..AND.ITN.GT.N) GO TO 24                             GG  88
      Z=3.*(F-FP)+GPDX+GDX                                              GG  89
      W=SQRT(1.-GDX/Z*GPDX/Z)*ABS(Z)                                    GG  90
      Z=1.-(GPDX+W-Z)/(DGDX+2.*W)                                       GG  91
      IF (Z.LT.0.1) Z=0.1                                               GG  92
      DO 11 I=1,N                                                       GG  93
      X(I)=X(I)-H(IDX+I)                                                GG  94
   11 CONTINUE                                                          GG  95
      GO TO 14                                                          GG  96
   12 F=FP                                                              GG  97
      DO 13 I=1,N                                                       GG  98
      G(I)=H(I)                                                         GG  99
   13 CONTINUE                                                          GG 100
      IF (DGDX.GT.0.) GO TO 15                                          GG 101
      GDX=GPDX                                                          GG 102
      Z=4.                                                              GG 103
   14 STEP=Z*STEP                                                       GG 104
      GO TO 8                                                           GG 105
   15 IF (GPDX.LT.0.5*GDX) STEP=2.*STEP                                 GG 106
      DGHDG=0.                                                          GG 107
      DO 19 I=1,N                                                       GG 108
      Z=0.                                                              GG 109
      IJ=IH+I                                                           GG 110
      IF (I.EQ.1) GO TO 17                                              GG 111
      II=I-1                                                            GG 112
      DO 16 J=1,II                                                      GG 113
      Z=Z+H(IJ)*H(IDG+J)                                                GG 114
      IJ=IJ+N-J                                                         GG 115
   16 CONTINUE                                                          GG 116
   17 DO 18 J=I,N                                                       GG 117
      Z=Z+H(IJ)*H(IDG+J)                                                GG 118
      IJ=IJ+1                                                           GG 119
   18 CONTINUE                                                          GG 120
      DGHDG=DGHDG+Z*H(IDG+I)                                            GG 121
      H(I)=Z                                                            GG 122
   19 CONTINUE                                                          GG 123
      IF (DGHDG.LT.0.0) DGHDG=DGDX*0.01                                 GG 124
      IF (DGDX.LT.DGHDG) GO TO 21                                       GG 125
      W=1.0+DGHDG/DGDX                                                  GG 126
      DO 20 I=1,N                                                       GG 127
      H(IDX+I)=W*H(IDX+I)-H(I)                                          GG 128
   20 CONTINUE                                                          GG 129
      DGDX=DGDX+DGHDG                                                   GG 130
      DGHDG=DGDX                                                        GG 131
   21 IJ=IH                                                             GG 132
      DO 22 I=1,N                                                       GG 133
      W=H(IDX+I)/DGDX                                                   GG 134
      Z=H(I)/DGHDG                                                      GG 135
      DO 22 J=I,N                                                       GG 136
      IJ=IJ+1                                                           GG 137
   22 H(IJ)=H(IJ)+W*H(IDX+J)-Z*H(J)                                     GG 138
      ITN=ITN+1                                                         GG 139
      GO TO 2                                                           GG 140
   23 IEXIT=5                                                           GG 141
   24 IF (IEXIT.EQ.1) KO=1                                              GG 142
      IF (IPRINT.EQ.0) RETURN                                           GG 143
      GO TO (25,26,27,26,28), IEXIT                                     GG 144
   25 WRITE (6,30) IEXIT                                                GG 145
      GO TO 29                                                          GG 146
   26 WRITE (6,31) IEXIT                                                GG 147
      GO TO 29                                                          GG 148
   27 WRITE (6,32) IEXIT                                                GG 149
      GO TO 29                                                          GG 150
   28 WRITE (6,33) IEXIT                                                GG 151
   29 CONTINUE                                                          GG 152
      RETURN                                                            GG 153
C                                                                       GG 154
   30 FORMAT (/,1HO,*IEXIT=*,I2,*CRITERION FOR OPTIMUM (CHANGE IN VECTOR GG 155
     1 X .LT.EPS) HAS BEEN SATISFIED*)                                  GG 156
   31 FORMAT (/,1HO,*IEXIT=*,I2,*EITHER OF THE FOLLOWING THINGS HAS HAPP GG 157
     1ENED*,/,9X,*1. EPS CHOSEN IS TOO SMALL*,/,9X,*2. GRADIENTS ARE NOT GG 158
     2 CORRECT*,/,9X,*3. MATRIX H GOES SINGULAR*)                       GG 159
   32 FORMAT (/,1HO,*IEXIT=*,I2,*MAXIMUM NUMBER OF ALLOWABLE ITERATION H GG 160
     1AS BEEN EXCEEDED*)                                                GG 161
   33 FORMAT (/,1HO,*IEXIT=*,I2,*FUNCTION VALUE LESS THAN MINIMUM ESTIMA GG 162
     1TE) HAS BEEN DETECTED*)                                           GG 163
      END                                                               GG 164
```

```
      SUBROUTINE FMFGG (FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H,IPRINT,GRAD,AP   H    1
     1P,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX,NP,INUMB,NINT,IP)            H    2
C                                                                          H    3
CC       PURPOSE                                                           H    4
CC          TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES     H    5
CC          ASSUMING THAT ITS GRADIENTS CAN BE CALCULATED EXPLICITLY       H    6
CC          BY THE METHOD OF FLETCHER AND POWELL                           H    7
CC                                                                         H    8
CC          THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE               H    9
CC          R. FLETCHER AND M.J.D. POWELL, A RAPIDLY CONVERGENT            H   10
CC          DESCENT METHOD FOR MINIMIZATION, COMP. JOURNAL,                H   11
CC          VOL.6, 1963, PP.163-168.                                       H   12
C                                                                          H   13
      COMMON /BLK/ KO                                                      H   14
      DIMENSION H(1), X(1), G(1), GRAD(1), NUMB(1), XX(3,1), XP(1), X1(1   H   15
     1), ERROR(1), EHELP(1), AP(1), INUMB(1)                               H   16
C                                                                          H   17
C        COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT   H   18
      KO=0                                                                 H   19
      CALL SECOND (T3)                                                     H   20
      CALL FUNCT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX   H   21
     1,NP,INUMB,NINT,IP)                                                   H   22
      KOUNT=0                                                              H   23
      NUMF=1                                                               H   24
      CALL SECOND (T4)                                                     H   25
      TIME=T4-T3                                                           H   26
      IF (IPRINT.EQ.0) GO TO 1                                             H   27
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)                                H   28
1     CONTINUE                                                             H   29
C                                                                          H   30
C        RESET ITERATION COUNTER AND GENERATE IDENTITY MATRIX              H   31
      IER=0                                                                H   32
      KK=0                                                                 H   33
      N2=N+N                                                               H   34
      N3=N2+N                                                              H   35
      N31=N3+1                                                             H   36
2     K=N31                                                                H   37
      DO 5 J=1,N                                                           H   38
      H(K)=1.                                                              H   39
      NJ=N-J                                                               H   40
      IF (NJ) 6,6,3                                                        H   41
3     DO 4 L=1,NJ                                                          H   42
      KL=K+L                                                               H   43
      H(KL)=0.                                                             H   44
4     CONTINUE                                                             H   45
      K=KL+1                                                               H   46
5     CONTINUE                                                             H   47
C                                                                          H   48
CC       START ITERATION LOOP                                              H   49
6     IF (KOUNT.EQ.0) GO TO 7                                              H   50
      IF (KK.NE.IPRINT) GO TO 7                                            H   51
      KK=0                                                                 H   52
      CALL SECOND (T4)                                                     H   53
      TIME=T4-T3                                                           H   54
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)                                H   55
7     CONTINUE                                                             H   56
      KOUNT=KOUNT+1                                                        H   57
      KK=KK+1                                                              H   58
C                                                                          H   59
C        SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR          H   60
      OLDF=F                                                               H   61
      DO 11 J=1,N                                                          H   62
      K=N+J                                                                H   63
      H(K)=G(J)                                                            H   64
      K=K+N                                                                H   65
      H(K)=X(J)                                                            H   66
C                                                                          H   67
C        DETERMINE DIRECTION VECTOR H                                      H   68
      K=J+N3                                                               H   69
      T=0.                                                                 H   70
      DO 10 L=1,N                                                          H   71
      T=T-G(L)*H(K)                                                        H   72
      IF (L-J) 8,9,9                                                       H   73
8     K=K+N-L                                                              H   74
      GO TO 10                                                             H   75
9     K=K+1                                                                H   76
10    CONTINUE                                                             H   77
      H(J)=T                                                               H   78
11    CONTINUE                                                             H   79
C                                                                          H   80
C        CHECK WHETHER FUNCTION WILL DECREASE STEPPING ALONG H.            H   81
      DY=0.                                                                H   82
      HNRM=0.                                                              H   83
      GNRM=0.                                                              H   84
C                                                                          H   85
CC       CALCULATE DIRECTIONAL DERIVATIVE AND TESTVALUES FOR DIRECTION     H   86
C        VECTOR H AND GRADIENT VECTOR G.                                   H   87
      DO 12 J=1,N                                                          H   88
      HNRM=HNRM+ABS(H(J))                                                  H   89
      GNRM=GNRM+ABS(G(J))                                                  H   90
      DY=DY+H(J)*G(J)                                                      H   91
```

```
12    CONTINUE                                                          H  92
C                                                                       H  93
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTIONAL  H  94
C     1  DERIVATIVE APPEARS TO BE POSITIVE OR ZERO.                     H  95
      IF (DY) 13,57,57                                                  H  96
C                                                                       H  97
CCC      REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION    H  98
C        VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.               H  99
13    IF (HNRM/GNRM-EPS) 57,57,14                                       H 100
CCC                                                                     H 101
CCC      SEARCH MINIMUM ALONG DIRECTION H                               H 102
CCC                                                                     H 103
CCC      SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE             H 104
14    FY=F                                                              H 105
      ALFA=2.*(EST-F)/DY                                                H 106
      AMBDA=1.                                                          H 107
C                                                                       H 108
CCC      USE ESTIMATE FOR STEPSIZE ONLY IF IT IS POSITIVE AND LESS THAN H 109
C        1. OTHERWISE TAKE 1. AS STEPSIZE                               H 110
      IF (ALFA) 17,17,15                                                H 111
15    IF (ALFA-AMBDA) 16,17,17                                          H 112
16    AMBDA=ALFA                                                        H 113
17    ALFA=0.                                                           H 114
CC                                                                      H 115
CC       SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT           H 116
18    FX=FY                                                             H 117
      DX=DY                                                             H 118
C                                                                       H 119
C        STEP ARGUMENT ALONG H                                          H 120
      DO 19 I=1,N                                                       H 121
      X(I)=X(I)+AMBDA*H(I)                                              H 122
19    CONTINUE                                                          H 123
CC                                                                      H 124
CC       COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT           H 125
      CALL FUNGT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX H 126
     1,NP,INUMB,NINT,IF)                                                H 127
      NUMF=NUMF+1                                                       H 128
      FY=F                                                              H 129
C                                                                       H 130
CC       COMPUTE DIRECTIONAL DERIVATIVE DY FOR NEW ARGUMENT.  TERMINATE H 131
CC       SEARCH, IF DY IS POSITIVE. IF DY IS ZERO THE MINIMUM IS FOUND  H 132
      DY=0.                                                             H 133
      DO 20 I=1,N                                                       H 134
      DY=DY+G(I)*H(I)                                                   H 135
20    CONTINUE                                                          H 136
      IF (DY) 21,41,24                                                  H 137
C                                                                       H 138
C        TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT     H 139
C        A MINIMUM HAS BEEN PASSED                                      H 140
21    IF (FY-FX) 22,24,24                                               H 141
CCC                                                                     H 142
CCC      REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES         H 143
22    AMBDA=AMBDA+ALFA                                                  H 144
      ALFA=AMBDA                                                        H 145
C        END OF SEARCH LOOP                                             H 146
CCC                                                                     H 147
C        TERMINATE IF THE CHANGE IN ARGUMENT GETS VERY LARGE            H 148
      IF (HNRM*AMBDA-1.E10) 18,18,23                                    H 149
C                                                                       H 150
C        LINEAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS       H 151
23    IER=2                                                             H 152
      GO TO 62                                                          H 153
C                                                                       H 154
CC       INTERPOLATE CUBICALLY IN THE INTERVAL DEFINED BY THE SEARCH    H 155
C        ABOVE AND COMPUTE THE ARGUMENT X FOR WHICH THE INTERPOLATION   H 156
C        POLYNOMIAL IS MINIMIZED                                        H 157
24    T=0.                                                              H 158
25    IF (AMBDA) 26,41,26                                               H 159
26    Z=3.*(FX-FY)/AMBDA+DX+DY                                          H 160
      ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))                                H 161
      DALFA=Z/ALFA                                                      H 162
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA                                 H 163
      IF (DALFA) 57,27,27                                               H 164
27    W=ALFA*SQRT(DALFA)                                                H 165
      ALFA=DY-DX+W+W                                                    H 166
      IF (ALFA) 28,29,28                                                H 167
28    ALFA=(DY-Z+W)/ALFA                                                H 168
      GO TO 31                                                          H 169
29    ALFA=(Z+DY-W)/(Z+DX+Z+DY)                                         H 170
30    ALFA=ALFA*AMBDA                                                   H 171
      DO 31 I=1,N                                                       H 172
      X(I)=X(I)+(T-ALFA)*H(I)                                           H 173
31    CONTINUE                                                          H 174
C                                                                       H 175
CC       TERMINATE, IF THE VALUE OF THE ACTUAL FUNCTION AT X IS LESS    H 176
CC       THAN THE FUNCTION VALUES AT THE INTERVAL ENDS. OTHERWISE REDUCE H 177
CC       THE INTERVAL BY CHOOSING ONE END-POINT EQUAL TO X AND REPEAT   H 178
CC       THE INTERPOLATION.  WHICH END-POINT IS CHOOSEN DEPENDS ON THE  H 179
CC       VALUE OF THE FUNCTION AND ITS GRADIENT AT X                    H 180
C                                                                       H 181
      NUMF=NUMF+1                                                       H 182
      CALL FUNGT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX H 183
     1,NP,INUMB,NINT,IP)                                                H 184
      IF (F-FX) 32,32,33                                                H 185
32    IF (F-FY) 41,41,33                                                H 186
33    DALFA=0.                                                          H 187
```

```
      DO 34 I=1,N                                                       H 188
      DALFA=DALFA+G(I)*H(I)                                             H 189
34    CONTINUE                                                          H 190
      IF (DALFA) 35,38,38                                               H 191
35    IF (F-FX) 37,36,38                                                H 192
36    IF (DX-DALFA) 37,41,37                                            H 193
37    FX=F                                                              H 194
      DX=DALFA                                                          H 195
      T=ALFA                                                            H 196
      AMBDA=ALFA                                                        H 197
      GO TO 25                                                          H 198
38    IF (FY-F) 40,39,40                                                H 199
39    IF (DY-DALFA) 40,41,40                                            H 200
40    FY=F                                                              H 201
      DY=DALFA                                                          H 202
      AMBDA=AMBDA-ALFA                                                  H 203
      GO TO 24                                                          H 204
C                                                                       H 205
C        TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION H 206
41    IF (OLDF-F+EPS) 57,42,42                                          H 207
C                                                                       H 208
C        COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM       H 209
C        TWO CONSECUTIVE ITERATIONS                                     H 210
42    DO 43 J=1,N                                                       H 211
      K=N+J                                                             H 212
      H(K)=G(J)-H(K)                                                    H 213
      K=N+K                                                             H 214
      H(K)=X(J)-H(K)                                                    H 215
43    CONTINUE                                                          H 216
C                                                                       H 217
C        TEST LENGTH OF ARGUMENT DIFFERENCE VECTOR AND DIRECTION VECTOR H 218
C        IF AT LEAST N ITERATIONS HAVE BEEN EXECUTED. TERMINATE, IF     H 219
C        BOTH ARE LESS THAN  EPS                                        H 220
      IER=1                                                             H 221
      IF (KOUNT-N) 47,44,44                                             H 222
44    T=0.                                                              H 223
      Z=0.                                                              H 224
      DO 45 J=1,N                                                       H 225
      K=N+J                                                             H 226
      W=H(K)                                                            H 227
      K=K+N                                                             H 228
      T=T+ABS(H(K))                                                     H 229
      Z=Z+W*H(K)                                                        H 230
45    CONTINUE                                                          H 231
      IF (HNRM-EPS) 46,46,47                                            H 232
46    IF (T-EPS) 62,62,47                                               H 233
C                                                                       H 234
C        TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED  LIMIT         H 235
47    IF (KOUNT-LIMIT) 48,55,55                                         H 236
C                                                                       H 237
C        PREPARE UPDATING OF MATRIX H                                   H 238
48    ALFA=0.                                                           H 239
      DO 52 J=1,N                                                       H 240
      K=J+N3                                                            H 241
      W=0.                                                              H 242
      DO 51 L=1,N                                                       H 243
      KL=N+L                                                            H 244
      W=W+H(KL)*H(K)                                                    H 245
      IF (L-J) 49,50,50                                                 H 246
49    K=K+N-L                                                           H 247
      GO TO 51                                                          H 248
50    K=K+1                                                             H 249
51    CONTINUE                                                          H 250
      K=N+J                                                             H 251
      ALFA=ALFA+W*H(K)                                                  H 252
      H(J)=W                                                            H 253
52    CONTINUE                                                          H 254
C                                                                       H 255
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS      H 256
C        ARE NOT SATISFACTORY                                           H 257
      IF (Z*ALFA) 53,2,53                                               H 258
C                                                                       H 259
C        UPDATE MATRIX H                                                H 260
53    K=N31                                                             H 261
      DO 54 L=1,N                                                       H 262
      KL=N2+L                                                           H 263
      DO 54 J=L,N                                                       H 264
      NJ=N2+J                                                           H 265
      H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA                            H 266
54    K=K+1                                                             H 267
      GO TO 6                                                           H 268
C        END OF ITERATION LOOP                                          H 269
C                                                                       H 270
C        NO CONVERGENCE AFTER  LIMIT  ITERATIONS                        H 271
55    IER=1                                                             H 272
      IF (KK.NE.IPRINT) GO TO 56                                        H 273
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT)                                  H 274
56    CONTINUE                                                          H 275
      GO TO 62                                                          H 276
C                                                                       H 277
C        RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS                   H 278
57    DO 58 J=1,N                                                       H 279
      K=N2+J                                                            H 280
      X(J)=H(K)                                                         H 281
58    CONTINUE                                                          H 282
      CALL FUNGT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,FHELP,AP,EMAX H 283
```

```
     1,NP,INUMB,NINT,IF)                                                    H 284
      NUMF=NUMF+1                                                           H 285
C                                                                           H 286
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE       H 287
C        FAILS TO BE SUFFICIENTLY SMALL                                     H 288
      IF (GNRM-EPS) 61,61,59                                                H 289
C                                                                           H 290
C        TEST FOR REPEATED FAILURE OF ITERATION                             H 291
59    IF (IER) 62,60,60                                                     H 292
60    IER=-1                                                                H 293
      GO TO 2                                                               H 294
61    IER=0                                                                 H 295
62    II=IER+2                                                              H 296
      IF (II.EQ.2) KO=1                                                     H 297
      IF (IPRINT.EQ.0) RETURN                                               H 298
      GO TO (63,64,65,66), II                                               H 299
63    WRITE (6,68) IER                                                      H 300
      GO TO 67                                                              H 301
64    WRITE (6,69) IER                                                      H 302
      GO TO 67                                                              H 303
65    WRITE (6,70) IER                                                      H 304
      GO TO 67                                                              H 305
66    WRITE (6,71) IER                                                      H 306
67    RETURN                                                                H 307
C                                                                           H 308
68    FORMAT (1HO,*IER=*,I2,* ERROR IN GRADIENTS CALCULATIONS*)             H 309
69    FORMAT (1HO,*IER=*,I2,* CRITERION FOR OPTIMUM HAS BEEN SATISFIED*)    H 310
70    FORMAT (1HO,*IER=*,I2,* MAXIMUM NUMBER OF ALLOWABLE ITERATIONS HAS    H 311
     1 BEEN EXCEEDED*)                                                      H 312
71    FORMAT (1HO,*IER=*,I2,* CHANGE IN ARGUMENTS GETS TOO LARGE, LINEAR    H 313
     1 SEARCH INDICATES THAT NO MINIMUM EXISTS*)                            H 314
      END                                                                   H 315-

HSBZ0CZ  //// END OF LIST ////
```

APPENDIX VII

Five section transmission line low pass filter

The $\underset{\sim}{A}$ matrix of a lossless transmission line is given by

$$\underset{\sim}{A} = \begin{bmatrix} \cos \beta\ell & jZ_0 \sin \beta\ell \\ \frac{j \sin \beta\ell}{Z_0} & \cos \beta\ell \end{bmatrix} \tag{7.1}$$

where Z_0 is the characteristic impedance of the transmission line, ℓ is its length and

$$\beta = \frac{2\pi f}{c}$$

where c is the velocity of propagation in the medium and f is frequency [37].

The reflection coefficient is given [37], from Fig. 7-1, as

$$\rho(j\omega) = 1 + 2R_g \frac{I_g(j\omega)}{V_g(j\omega)} . \tag{7.2}$$

The problem defined in Test 3, Chapter VI is to optimize the absolute value of the reflection coefficient (7.2) for the low pass filter design. The impedances $R_g = R_L = 1\Omega$.

To compute the reflection coefficient and its gradients it is easier to assume that the load current is 1 instead of fixing the value of the input generator V_g[37]. In this case

$$\begin{bmatrix} V_L \\ I_L \end{bmatrix} \underset{\sim}{\Delta} \begin{bmatrix} V_6 \\ I_6 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} .$$

Using the relation (7.1) we can write

$$\begin{bmatrix} V_i \\ I_i \end{bmatrix} = \begin{bmatrix} \cos\beta\ell_i & jZ_{oi}\sin\beta\ell_i \\ \frac{j\sin\beta\ell_i}{Z_{oi}} & \cos\beta\ell_i \end{bmatrix} \begin{bmatrix} V_{i+1} \\ I_{i+1} \end{bmatrix} \quad , \; i=5,4,3,2,1.$$

Also from Fig. 7-1

$$I_g = -I_1$$

$$V_g = V_1 - I_g R_g.$$

The adjoint network approach was used to calculate the gradients

$$\nabla\left(\frac{I_g}{V_g}\right) = \frac{1}{V_g^2}\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \end{bmatrix}$$

where V_g is not a function of design variables and

$$g_i = \frac{1}{Z_{oi}}\left(V_i I_i - V_{i+1} I_{i+1}\right), \qquad i=1,2,3,4,5$$

are the sensitivities which may be found in tables in [37].

The gradients of the $|\rho|$ with respect to the variable characteristic impedances are given by

$$\nabla|\rho| = 2R_g \cdot \mathrm{Re}\left\{\frac{|\rho|}{\rho} \cdot \nabla\left(\frac{Ig}{V_g}\right)\right\}.$$

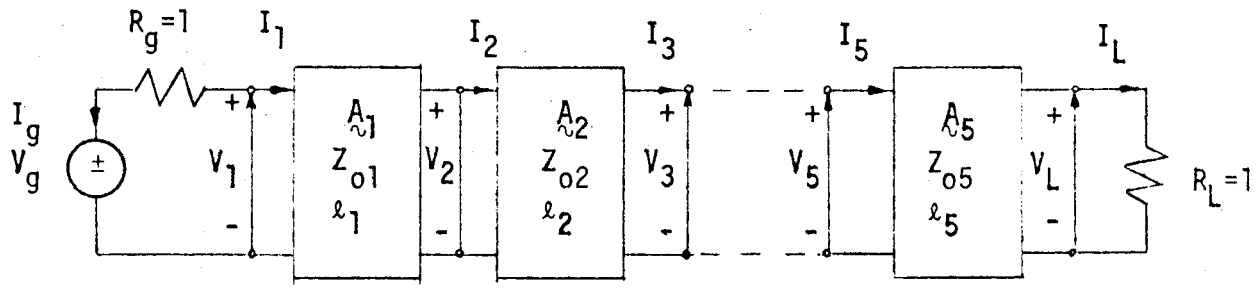


Fig. 7-1 Cascaded five section transmission-line network.

REFERENCES

[1] J.W. Bandler and C. Charalambous, "Practical least pth approximation with extremely large values of p", in Proc. 5th Asilomar Conf. on Circuits and Systems, (Pacific Grove, Calif. 1971), pp. 66-70.

[2] J.W. Bandler and C. Charalambous, "Theory of generalized least pth approximation", IEEE Trans. Circuit Theory, Vol. CT-19, pp. 287-289, 1972.

[3] G.C. Temes and D.A. Calahan, "Computer-aided network optimization, the state-of-the-art", Proc. IEEE, Vol. 55, pp. 1832-1863, 1967.

[4] M.J. Box, "A comparison of several current optimization methods, and the use of transformations in constrained problems", Comp. J., Vol. 9, pp. 67-77, 1966.

[5] A.V. Fiacco and G.P. McCormick, "The sequential unconstrained minimization technique for nonlinear programming, a primal-dual method", Management Sci., Vol. 10, pp. 360-366, 1964.

[6] A.V. Fiacco and G.P. McCormick, "Computational algorithm for the sequential unconstrained minimization technique for nonlinear programming", Management Sci., Vol. 10, pp. 601-617, 1964.

[7] A.D. Waren, L.S. Lasdon, and D.F. Suchman, "Optimization in engineering design", Proc. IEEE, Vol. 55, pp. 1885-1897, 1967.

[8] W.I. Zangwill, Nonlinear Programming: A Unified Approach, Englewood Cliffs, N.J.: Prentice Hall, 1969.

[9] L.S. Lasdon and A.D. Waren, "Mathematical programming for

optimal design", Electro-Technol., Vol. 80, pp. 55-70, 1967.

[10] D.J. Wilde, Optimum Seeking Methods, Englewood Cliffs, N.J.: Prentice Hall, 1964.

[11] D.J. Wilde and C.S. Beightler, Foundations of Optimization, Englewood Cliffs, N.J.: Prentice Hall, 1967.

[12] J. Kowalik and M.R. Osborne, Methods for Unconstrained Optimization Problems, Elsevier Publishing Co. Inc., New York, 1968.

[13] M.J. Box, D. Davies and W.H. Swann, Nonlinear Optimization Techniques, Oliver and Boyd Ltd., Edinburgh, 1969.

[14] W.C. Davidon, "Variable metric method for minimization", A.E.C. Research and Development Report, ANL-5990 (Rev.), 1959.

[15] A.S. Householder, Principles of Numerical Analysis, McGraw-Hill, New York, 1953.

[16] C.G. Broyden, "Quasi-Newton method and their application to function minimization", Math. of Comp., Vol. 21, pp. 368-381, 1967.

[17] H.Y. Huang, "Unified approach to quadratically convergent algorithms for function minimization", J. Optimization Theory and Applications, Vol. 5, pp. 405-423, 1970.

[18] R. Fletcher and M.J.D. Powell, "A rapidly convergent descent method for minimization", Comp. J., Vol. 6, pp. 163-168, 1963.

[19] R. Fletcher, "A new approach to variable metric algorithms," Comp. J., Vol. 13, pp. 317-322, 1970.

[20] A. Ralston, "Rational Chebyshev approximation by Remes'

algorithms", Numer, Math., No. 7, pp. 322-331, 1965.

[21] W.J. Cody and J. Stoer, "Rational Chebyshev approximation using interpolation", Numer. Math., No. 9, pp. 177-188, 1966.

[22] H. Werner, J. Stoer and W. Bommas, "Rational Chebyshev approximation", Numer. Math., No. 10, pp. 289-306, 1967.

[23] W.J. Cody, "A survey of practical rational and polynomial approximation of functions", SIAM Rev., Vol. 12, pp. 401-423, 1970.

[24] M.R. Aaron, "The use of least squares in system design", IRE Trans. Circuit Theory, Vol. CT-3, pp. 224-231, 1956.

[25] D.W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters", J. SIAM, Vol. 11, pp. 431-441, 1963.

[26] G.C. Temes, "Optimization methods in circuit design", in Computer Oriented Circuit Design, F.F. Kuo and W.G. Magnusson, Jr., Eds. Englewood Cliffs, N.J.: Prentice Hall, 1969.

[27] G.C. Temes and D.Y.F. Zai, "Least pth approximation", IEEE Trans. Circuit Theory, Vol. CT-16, pp. 235-237, 1969.

[28] J.W. Bandler and C. Charalambous, "On conditions for optimality in least pth approximation with $p \to \infty$", in Proc. 9th Allerton Conf. on Circuit and System Theory, (Urbana, Ill., 1971), pp. 404-413. J. Optimization Theory and Applications, accepted for publication.

[29] J.W. Bandler, "Conditions for a minimax optimum", IEEE Trans. Circuit Theory, Vol. CT-18, pp. 476-479, 1971.

[30] N.I. Achieser, Theory of Approximation (Translated by C.J. Hyman), Frederick Ungar Publ. Co., New York, 1956.

[31] L. Fox and I.B. Parker, Chebyshev Polynomials in Numerical Analysis, Oxford University Press, 1968.

[32] J.R. Rice, "The characterization of best nonlinear Tchebycheff approximations", Trans. Amer. Math. Soc., Vol. 96, pp. 322-340, 1960.

[33] A.R. Curtis and M.J.D. Powell, "Necessary conditions for a minimax approximation", Comp. J., Vol. 8, pp. 358-361, 1965.

[34] J.W. Bandler and C. Charalambous, "Practical least pth optimization of networks", IEEE Trans. Microwave Theory Tech., Vol. MTT-20, 1972.

[35] H.J. Carlin, "Distributed Circuit Design With Transmission Line Elements", Proc. IEEE, Vol. 59, 1059-1081, 1971.

[36] J.W. Bandler and V.K. Jha, "GRADMIN- a package for function minimization using efficient gradient methods", Department of Electrical Engineering, McMaster Univ., Hamilton, Ont., Canada, 1972, internal report.

[37] J.W. Bandler and R.E. Seviora, "Current trends in network optimization", IEEE Trans. Microwave Theory Tech., Vol. MTT-18, pp. 1159-1170, 1970.