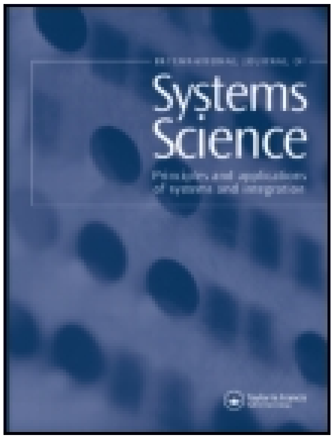


International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tsys20

General programmes for least pth and near minimax approximation

J. R. POPOVIĆ [a], J. W. BANDLER [a] & C. CHARALAMBOUS [b]

[a] Department of Electrical Engineering, McMaster University, Hamilton, Ontario, Canada

[b] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada

Published online: 27 Mar 2007.

INT. J. SYSTEMS SCI., 1974, VOL. 5, NO. 10, 907–932

General programmes for least *p*th and near minimax approximation

J. R. POPOVIĆ, J. W. BANDLER
Department of Electrical Engineering, McMaster University, Hamilton, Ontario, Canada

and C. CHARALAMBOUS
Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada

[Received 20 February 1973 ; revised 30 April 1973]

User-oriented computer programmes 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 non-linear 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 method due to Fletcher. It has been demonstrated how efficiently extremely near minimax results can be achieved on a discrete set of sample points using this approach and the programmes written verify this. The programmes 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.

1. Introduction

Two complete user-oriented computer programmes in FORTRAN IV are presented which utilize some new ideas on discrete least pth approximation (Bandler and Charalambous 1972). 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 (1963) and, more recently, to Fletcher (1970) are used. The user has to write all the required specifications, the approximating functions and weighting functions in a straightforward way.

The first programme is described in § 4 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 minimax approximation problem. Numerical examples for which the minimax solutions are known were chosen to illustrate the work of the programme. The solutions obtained are in excellent agreement with the known ones.

The second programme is described in § 6. The programme 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 programme works were chosen in system modelling and multi-section loss-less transmission line network design. Although the programme is not written for non-linear programming it is also applicable for problems with parameter constraints.

2. Definitions

Define real weighted error functions related to the upper and lower specifications, respectively, as (Bandler and Charalambous 1972) :

$$e_u(\mathbf{a}, x) \triangleq w_u(x)(F(\mathbf{a}, x) - S_u(x)), \tag{1}$$

$$e_u'(\mathbf{a}, x, \xi) \triangleq w_u(x)(F(\mathbf{a}, x) - S_u'(x, \xi)) = e_u(\mathbf{a}, x) - \xi, \tag{2}$$

$$e_l(\mathbf{a}, x) \triangleq w_l(x)(F(\mathbf{a}, x) - S_l(x)), \tag{3}$$

$$e_l'(\mathbf{a}, x, \xi) \triangleq w_l(x)(F(\mathbf{a}, x) - S_l'(x, \xi)) = e_l(\mathbf{a}, x) + \xi, \tag{4}$$

where the symbols are

- $F(\mathbf{a}, x)$ approximating function,
- $S_u(x)$ upper specified function,
- $S_u'(x, \xi)$ artificial upper specified function,
- $S_l(x)$ lower specified function,
- $S_l'(x, \xi)$ artificial lower specified function,
- $w_u(x)$ upper positive weighting function,
- $w_l(x)$ lower positive weighting function,
- $\mathbf{a}$ vector containing the k independent parameters,
- x independent variable,
- ξ margin of errors with respect to the artificial and desired specifications.

When upper and lower specified functions and weighting functions coincide, respectively, let

$$S(x) = S_u(x) = S_l(x), \tag{5}$$

$$w(x) = w_u(x) = w_l(x), \tag{6}$$

then from (1) and (3)

$$e(\mathbf{a}, x) = e_u(\mathbf{a}, x) = e_l(\mathbf{a}, x). \tag{7}$$

In practice we will evaluate all the functions at a finite discrete set of values of x taken from one or more closed intervals. Therefore, we will let

$$e_{ui}'(\mathbf{a}, \xi) \triangleq e_u'(\mathbf{a}, x_i, \xi), \quad i \in I_u, \tag{8}$$

$$e_{li}'(\mathbf{a}, \xi) \triangleq e_l'(\mathbf{a}, x_i, \xi), \quad i \in I_l, \tag{9}$$

$$e_i(\mathbf{a}) \triangleq e(\mathbf{a}, x_i), \qquad i \in I_s, \tag{10}$$

where it is assumed that a sufficient number of sample points have been chosen so that the discrete approximation problem adequately approximates the continuous problem. I_u, I_l and I_s are appropriate index sets.

The artificial margin ξ allows for certain flexibility in formulating the optimization problem, and will be discussed at a later stage.

3. Background theory

Consider a system of real non-linear functions,

$$f_i(\mathbf{a}, \xi) \triangleq e_{ui}'(\mathbf{a}, \xi), \qquad i \in I_u, \tag{11}$$

$$f_i(\mathbf{a}, \xi) \triangleq -e_{li}'(\mathbf{a}, \xi), \quad i \in I_l, \tag{12}$$

Bandler and Charalambous (1972) proposed the generalized least pth objective function which is valid for both negative and non-negative f_i for $i \in I \triangleq I_u \cup I_l$ and which alleviates the ill-conditioning resulting from the numerical evaluation of $[\pm f_i(\mathbf{a}, \xi)]^{\pm p}$ for very large values of p, namely,

$$U(\mathbf{a}, \xi) = M(\mathbf{a}, \xi) \left(\sum_{i \in K} \left[\frac{f_i(\mathbf{a}, \xi)}{M(\mathbf{a}, \xi)} \right]^q \right)^{1/q} \quad \text{for} \quad M(\mathbf{a}, \xi) \neq 0, \tag{13}$$

where

$$M(\mathbf{a}, \xi) \triangleq \max_{i \in I} f_i(\mathbf{a}, \xi), \tag{14}$$

$$q \triangleq \frac{M(\mathbf{a}, \xi)}{|M(\mathbf{a}, \xi)|} \cdot p \begin{Bmatrix} p > 1 & \text{if} & M(\mathbf{a}, \xi) > 0 \\ p \geqslant 1 & \text{if} & M(\mathbf{a}, \xi) < 0 \end{Bmatrix} \tag{15}$$

and

$$K \triangleq \begin{Bmatrix} J \triangleq \{ i \mid f_i(\mathbf{a}, \xi) \geqslant 0,\ i \in I \} & \text{if} & M(\mathbf{a}, \xi) > 0 \\ I & \text{if} & M(\mathbf{a}, \xi) < 0 \end{Bmatrix} \tag{16}$$

The gradients of the objective function (13) are

$$\nabla U(\mathbf{a}, \xi) = \left(\sum_{i \in K} \left[\frac{f_i(\mathbf{a}, \xi)}{M(\mathbf{a}, \xi)} \right]^q \right)^{(1/q)-1} \cdot \left(\sum_{i \in K} \left[\frac{f_i(\mathbf{a}, \xi)}{M(\mathbf{a}, \xi)} \right]^{q-1} \nabla f_i(\mathbf{a}, \xi) \right) \tag{17}$$

where

$$\nabla \triangleq \left[\frac{\partial}{\partial a_1} \frac{\partial}{\partial a_2} \cdots \frac{\partial}{\partial a_k} \right]^{\mathrm{T}}.$$

By minimizing the objective function defined by (13) with a large value of p we should obtain results very close to the minimax optimum (Bandler and Charalambous 1973).

If $\xi = 0$, $f_i > 0$ indicates that a specification or a response constraint is violated, and $f_i < 0$ that a specification is exceeded ; $f_i = 0$ indicates that a specification is met exactly. It is quite possible that some of the f_i are equal to $-\infty$ in which case they are simply ignored by (13). Also the generalized objective does not allow any of the f_i to be $+\infty$. If the $f_i(\mathbf{a}, \xi)$ for $i \in I$ are continuous with continuous partial derivatives, the proposed objective function is continuous with continuous partial derivatives. The objective function (13) and partial derivatives (17) still remain continuous even when, for some i's, the f_i are discontinuous or continuous with discontinuous derivatives, simply because those points are ignored.

The 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 $M(\mathbf{a}, \xi)$ of (14) is always positive or negative during optimization. When $M(\mathbf{a}, \xi)$ is positive, only sample points which belong to index set J (16) are considered and, therefore, there is a saving in gradient computation. But in this case it may happen that $M(\mathbf{a}, \xi) = 0$, when the function (13) is continuous but the derivatives may be 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 $M(\mathbf{a}, \xi)$ is chosen to be negative this possible failure is avoided.

4. The computer programme FMCLP

We will consider first the programme written for minimizing the objective function corresponding to a single specified function. A function f_i is chosen to be the absolute value of a single specified weighted error function (10) for all $i \in I_s$. To alleviate the ill-conditioning for very large values of p, a similar scaling as in (14) was proposed (Bandler and Charalambous 1971).

$$U(\mathbf{a}) = M(\mathbf{a}) \left(\sum_{i \in I_s} \left| \frac{e_i(\mathbf{a})}{M(\mathbf{a})} \right|^p \right)^{1/p} \quad \text{for} \quad 1 < p < \infty, \tag{18}$$

where

$$M(\mathbf{a}) \triangleq \max_{i \in I_s} |e_i(\mathbf{a})|. \tag{19}$$

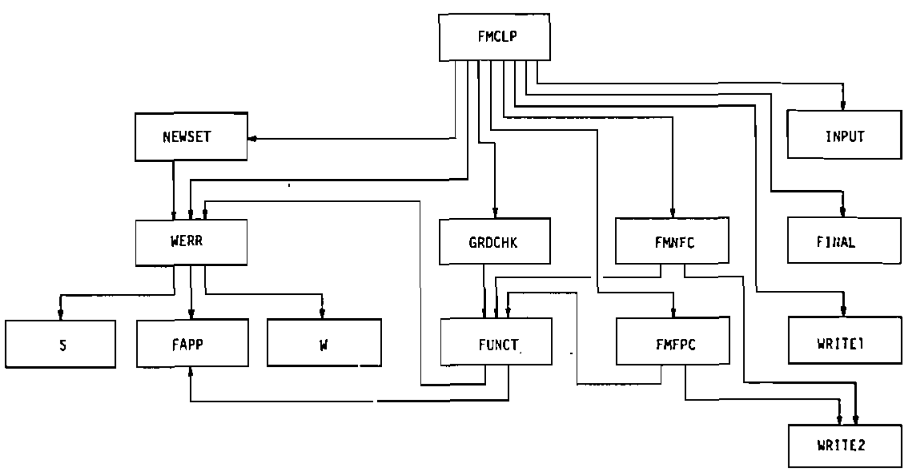


Figure. 1 The organization of FMCLP.

A list and a brief description of the 14 sub-programmes comprising FMCLP is given below :

FMCLP Supplies data for the function minimization and coordinates the other sub-programmes (see figs. 1 and 2).

S Defines a specified function (5) over an interval.

FAPP Defines an approximating function over an interval and the gradients with respect to variable parameters.

W Defines a weighting function (6) over an interval.

WERR The output of this sub-programme has a value of the weighted error (10) at a single point x for a particular vector $\mathbf{a}$.

NEWSET Redefines a sample point set such as to include all the extreme points in the summation of the objective function (18). Quadratic interpolation is used to locate the extreme points more precisely (see § 5 and fig. 3).

FUNCT Keeps the values of the weighted error of each sample point in an array, finds the maximum absolute value and computes the objective function (18) and its gradients.

GRDCHK Checks the gradients with respect to all variable parameters before the optimization process starts by testing,

$$\left| \frac{\dfrac{U(a_i+\Delta a_i)-U(a_i)}{\Delta a_i} - \dfrac{\partial U}{\partial a_i}}{\dfrac{U(a_i+\Delta a_i)-U(a_i)}{\Delta a_i}} \right| < \eta, \quad i=1, \ldots, k. \tag{20}$$

FMFPC Minimizes a function using the Fletcher–Powell method.

FMNFC Minimizes a function using the Fletcher method.

INPUT Prints the input data for the optimization process.

FINAL Prints the optimum solution.

WRITE 1 and WRITE 2 Print the intermediate results, if desired.

S, W and WERR are function sub-programmes and the others are subroutine sub-programmes.

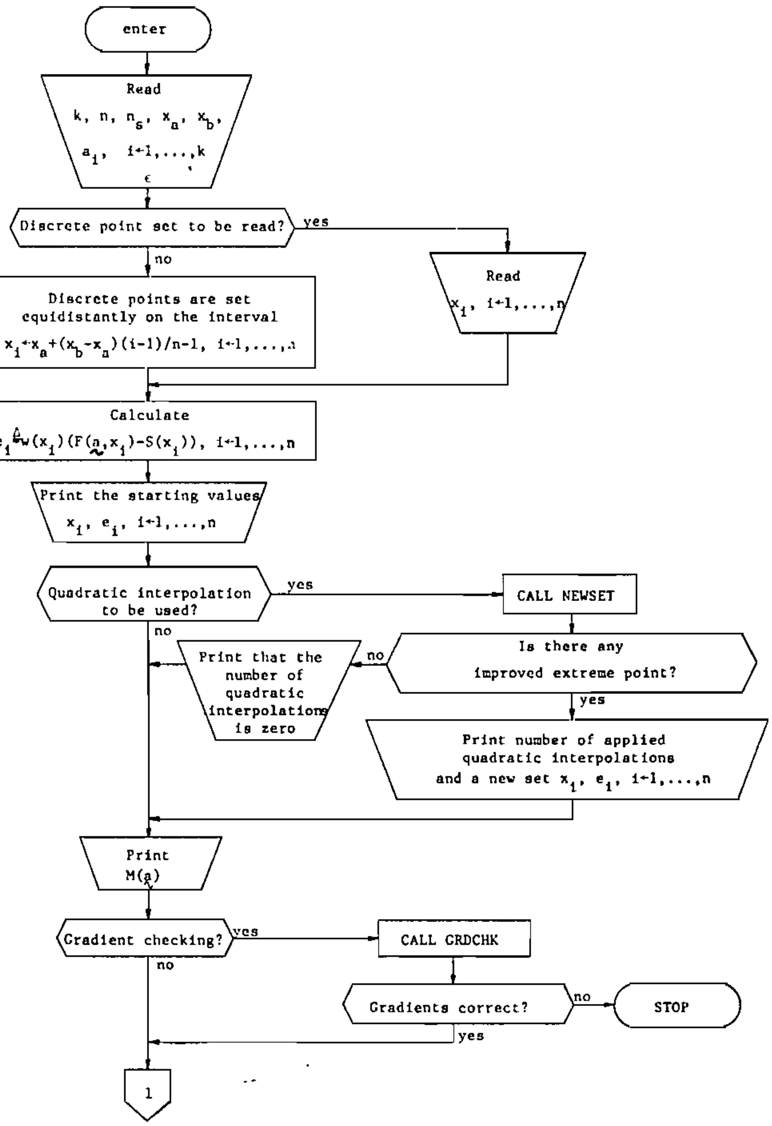
A user of FMCLP writes S, FAPP and W.

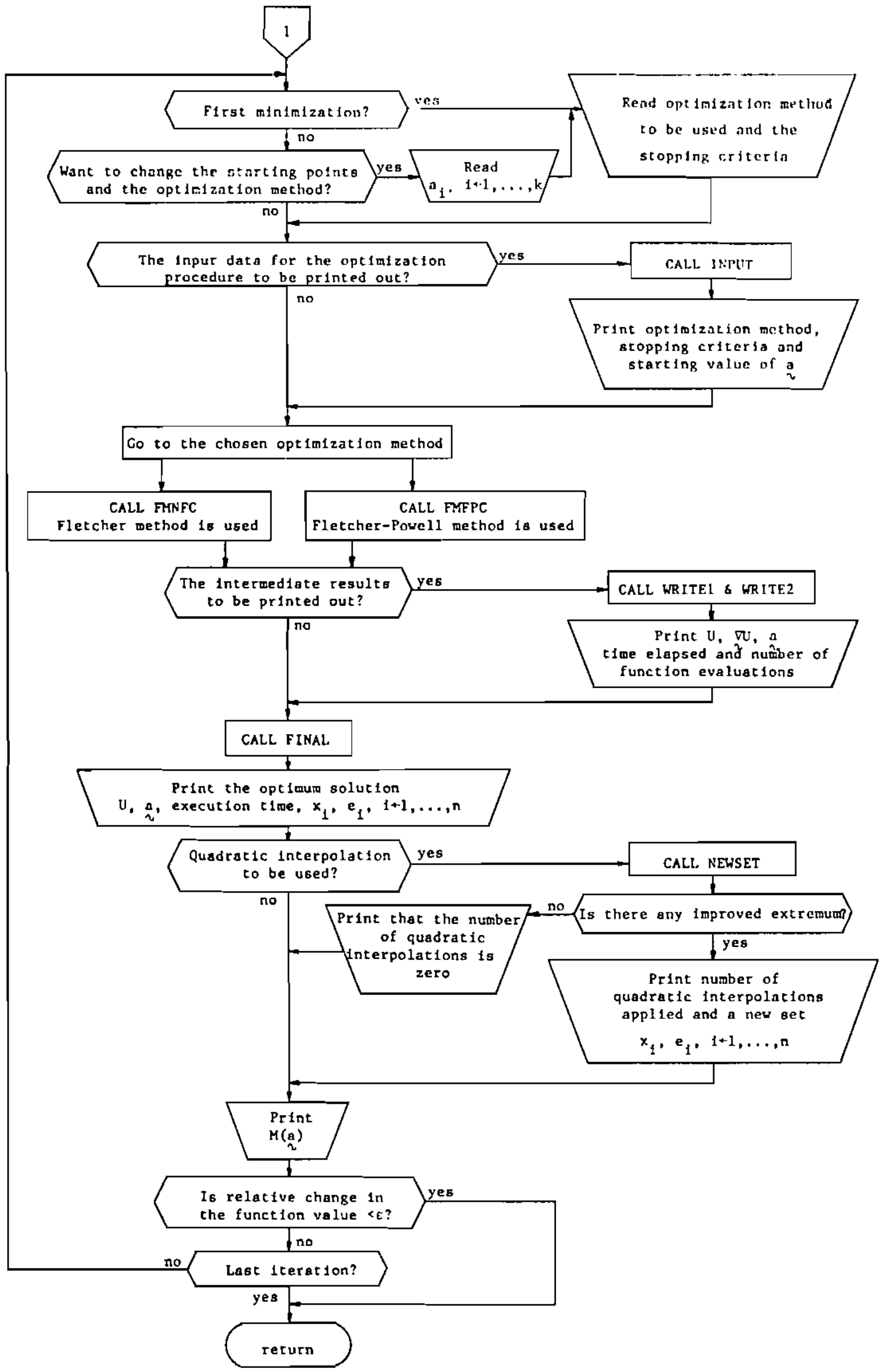
The programme terminates when stopping criteria for the Fletcher–Powell or Fletcher method are satisfied or when the relative change in the objective function in two successive iterations is less than a small prescribed quantity ϵ.

5. Quadratic interpolation

If the requirement is a minimax approximation 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. Retaining the maxima and

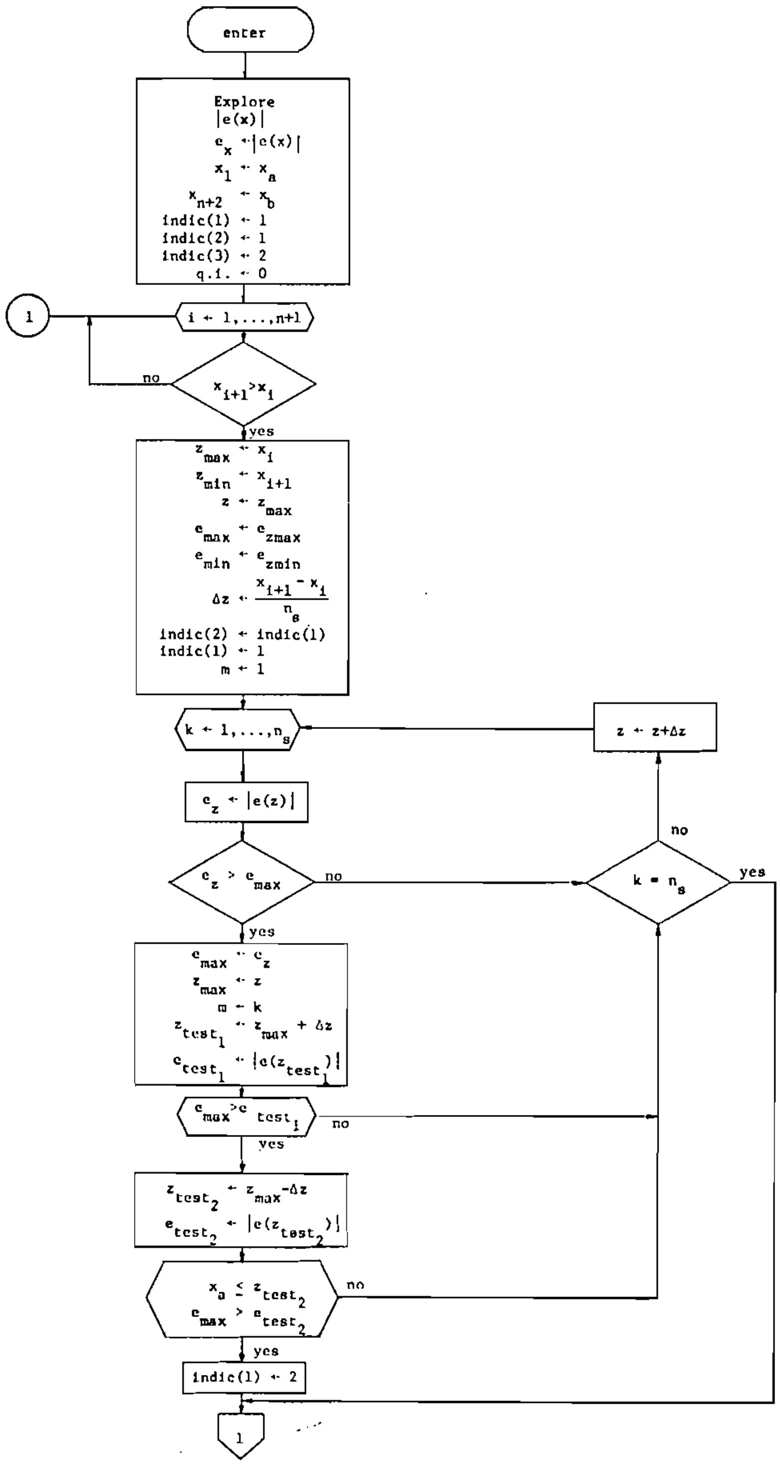
Fig. 2

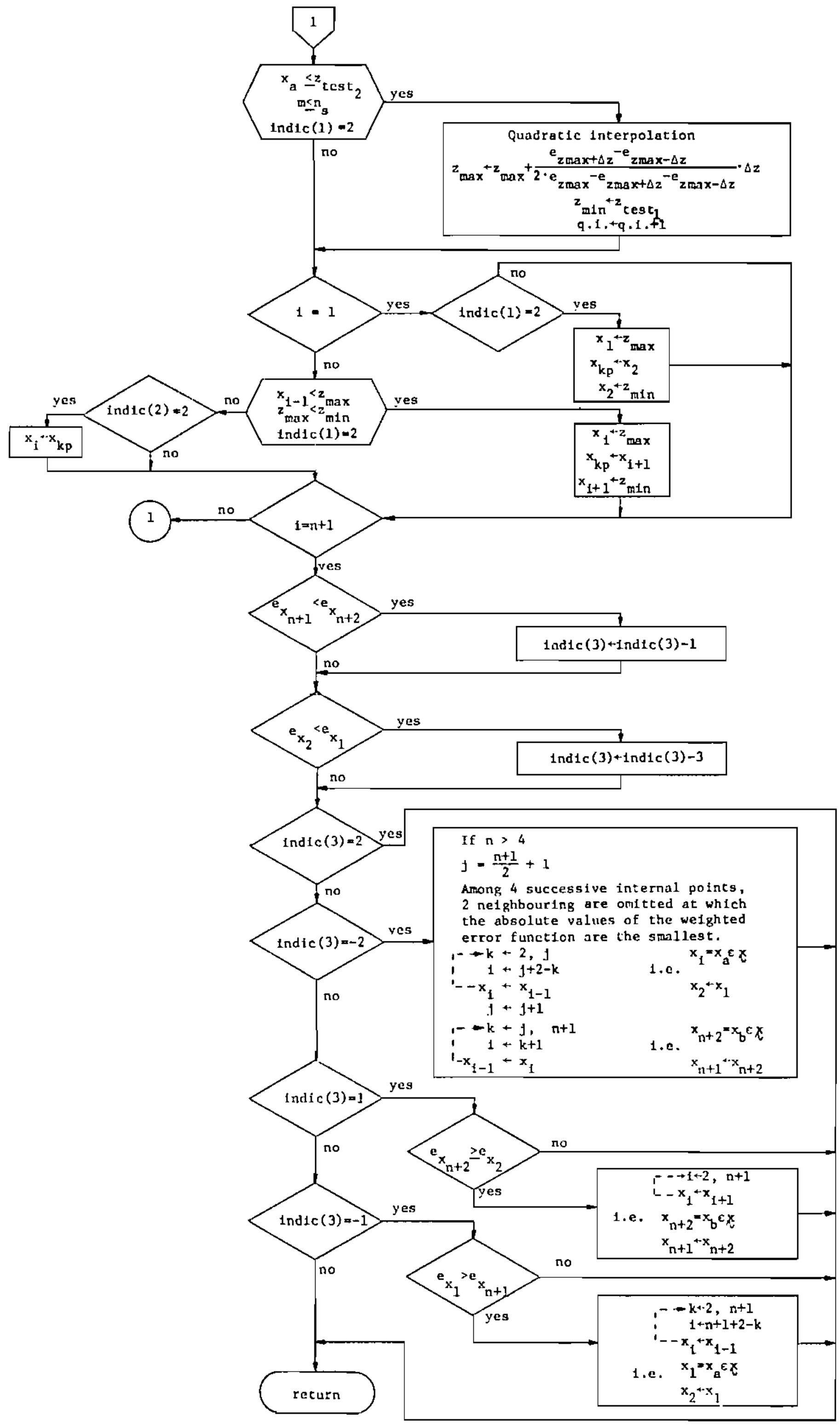

enter
Read
k, n, n_s, x_a, x_b,
a_i, i←1,...,k
ε
Discrete point set to be read?
yes
no
Read
x_i, i←1,...,n
Discrete points are set
equidistantly on the interval
x_i←x_a+(x_b-x_a)(i-1)/n-1, i←1,...,n
Calculate
e_i ≜ w(x_i)(F(a,x_i)-S(x_i)), i←1,...,n
Print the starting values
x_i, e_i, i←1,...,n
Quadratic interpolation
to be used?
yes
CALL NEWSET
no
Print that the
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, e_i, i←1,...,n
Print
M(a)
Gradient checking?
yes
CALL GRDCHK
no
Gradients correct?
no
STOP
yes
1



Flowchart of subroutine FMCLP.

Fig. 3


enter
Explore |e(x)|
e_x ← |e(x)|
x_1 ← x_a
x_{n+2} ← 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 ← x_i
z_min ← x_{i+1}
z ← z_max
e_max ← e_zmax
e_min ← e_zmin
Δz ← (x_{i+1} − x_i)/n_s
indic(2) ← indic(1)
indic(1) ← 1
m ← 1
k ← 1,...,n_s
z ← z+Δz
e_z ← |e(z)|
no
e_z > e_max
no
k = n_s
yes
yes
e_max ← e_z
z_max ← z
m ← k
z_test_1 ← z_max + Δz
e_test_1 ← |e(z_test_1)|
e_max > e_test_1
no
yes
z_test_2 ← z_max − Δz
e_test_2 ← |e(z_test_2)|
x_a ≤ z_test_2
e_max > e_test_2
no
yes
indic(1) ← 2
1



Flowchart of subroutine NEWSET (see Appendix for definitions of some parameters).

removing from the objective function those sample points which do not substantially contribute to the summation may save computation time. Even more can be done if approximations to the actual maxima replace the sample points in their neighbourhood.

It is assumed that, in the neighbourhood of an extremum, the function is adequately represented 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 obtained. This point replaces one of the initial points.

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

$$\{x_i^{(j)}\}, \quad i \in I_s \tag{21}$$

be the set of n sample points at the beginning of the jth iteration. Before the searching procedure for the extrema of weighted error starts, the end points of the interval are added to the given set (21), 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 $x_n \neq x_b$) sub-intervals over $[x_a, x_b]$. Each sub-interval is divided by a predicted n_s equidistant grid of points. Let

$$\{z_k^{(i,j)}\}, \quad k = 1, 2, \ldots, n_s + 1 \tag{22}$$

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 sub-interval obtained up to that time. If both neighbouring points have absolute values of the weighted error less than the current one, then the extremum $\hat{z}_k^{(i,j)}$ is found by applying the quadratic interpolation. This point replaces $x_i^{(j)}$, the left end point of the current sub-interval.

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 sub-interval. This is done in case there are more than one extremum on a single sub-interval. Thus the other possible extrema are 'removed' to the next sub-interval by removing its left end point. However, if there are no extreme points on the next sub-interval, the end point is again set to its previous value.

All the extrema are selected by applying this searching procedure on every sub-interval and these points replace the nearby left hand side point obtained up to that time, and the new set $\{x_i^{(j+1)}\}$, $i \in I_s$ 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.

In an effort to keep the number of the discrete points in the summation (18) constant we select n sample points from $n+2$ 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 large errors, two neighbouring points are omitted from four successive internal points where the absolute values of the weighted error function are the smallest.

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.

A flowchart of this procedure is illustrated in fig. 3.

6. The computer programme FMLPO

Here, we will consider a programme written for generalized least pth approximation described in § 3. A list and a brief description of the 15 sub-programmes comprising FMLPO is given below:

- FMLPO Supplies data for the optimization process and coordinates the other sub-programmes (see figs. 4 and 5).
- 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.
- EPSNP Computes the upper or lower weighted error at a single point x in a particular interval and for a particular vector $\mathbf{a}$.
- ERRO Selects weighted error functions according to (16) (see fig. 6).
- FUNGT Computes the generalized least pth objective function (13) and its gradients (17) (see fig. 7).

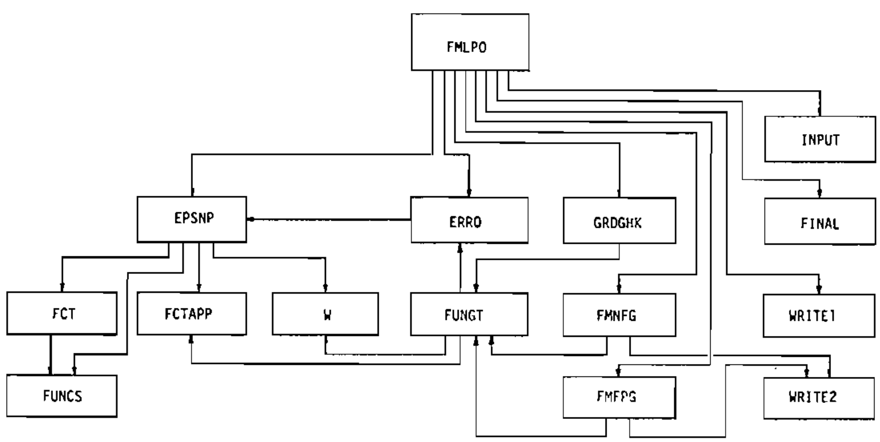
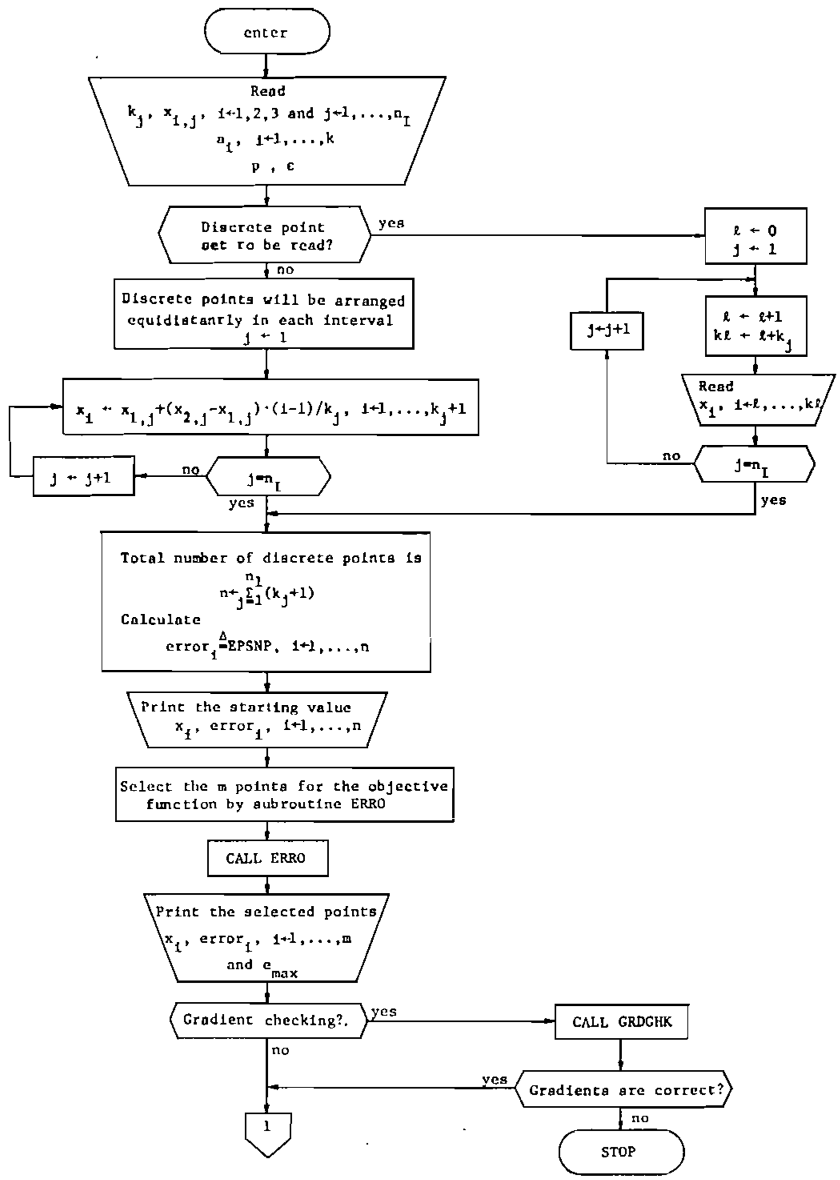


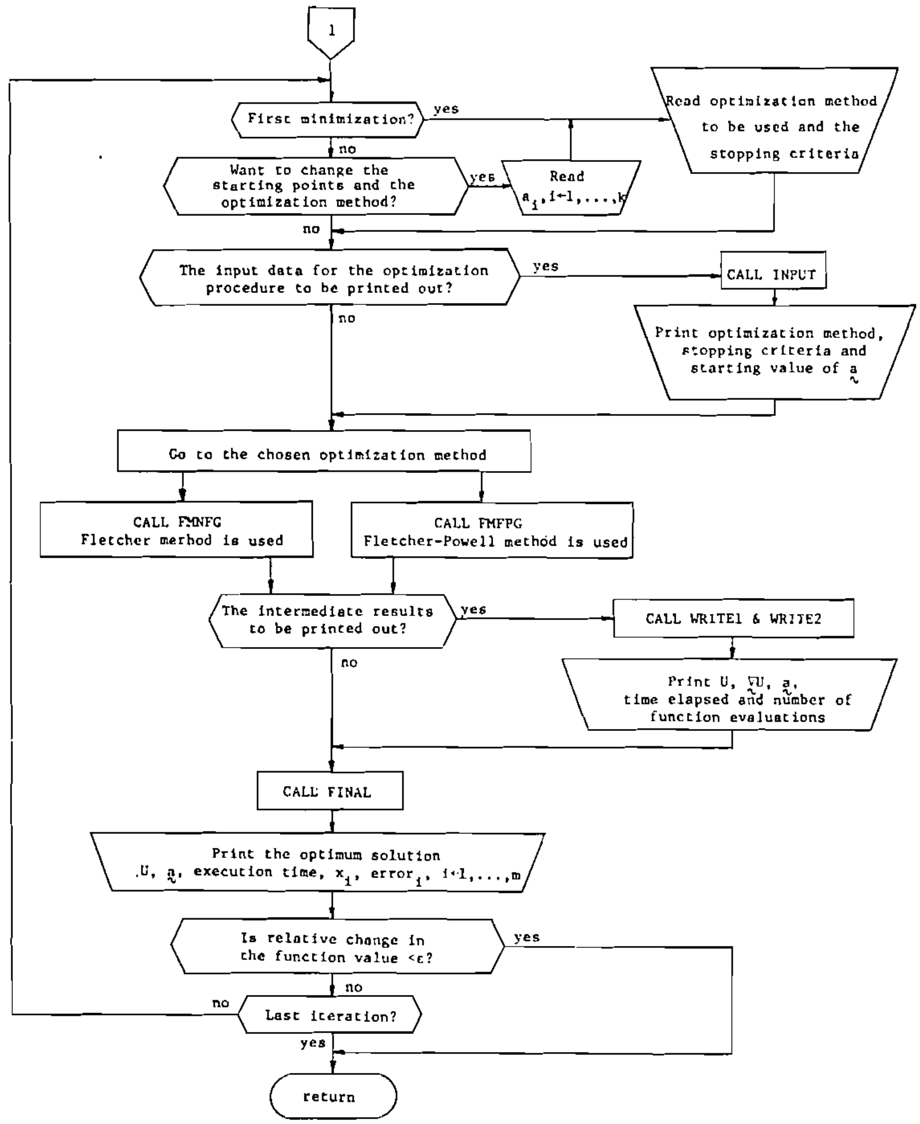
Figure. 4 The organization of FMLPO.

GRDGHK, FMFPG and FMNFG are subroutine sub-programmes which have the same role as the subroutines GRDCHK, FMFPC and FMNFC, respectively, in the FMCLP package. Different names are for convenience only.

INPUT, FINAL, WRITE 1 and WRITE 2 are subroutine sub-programmes which have already been introduced in § 4.

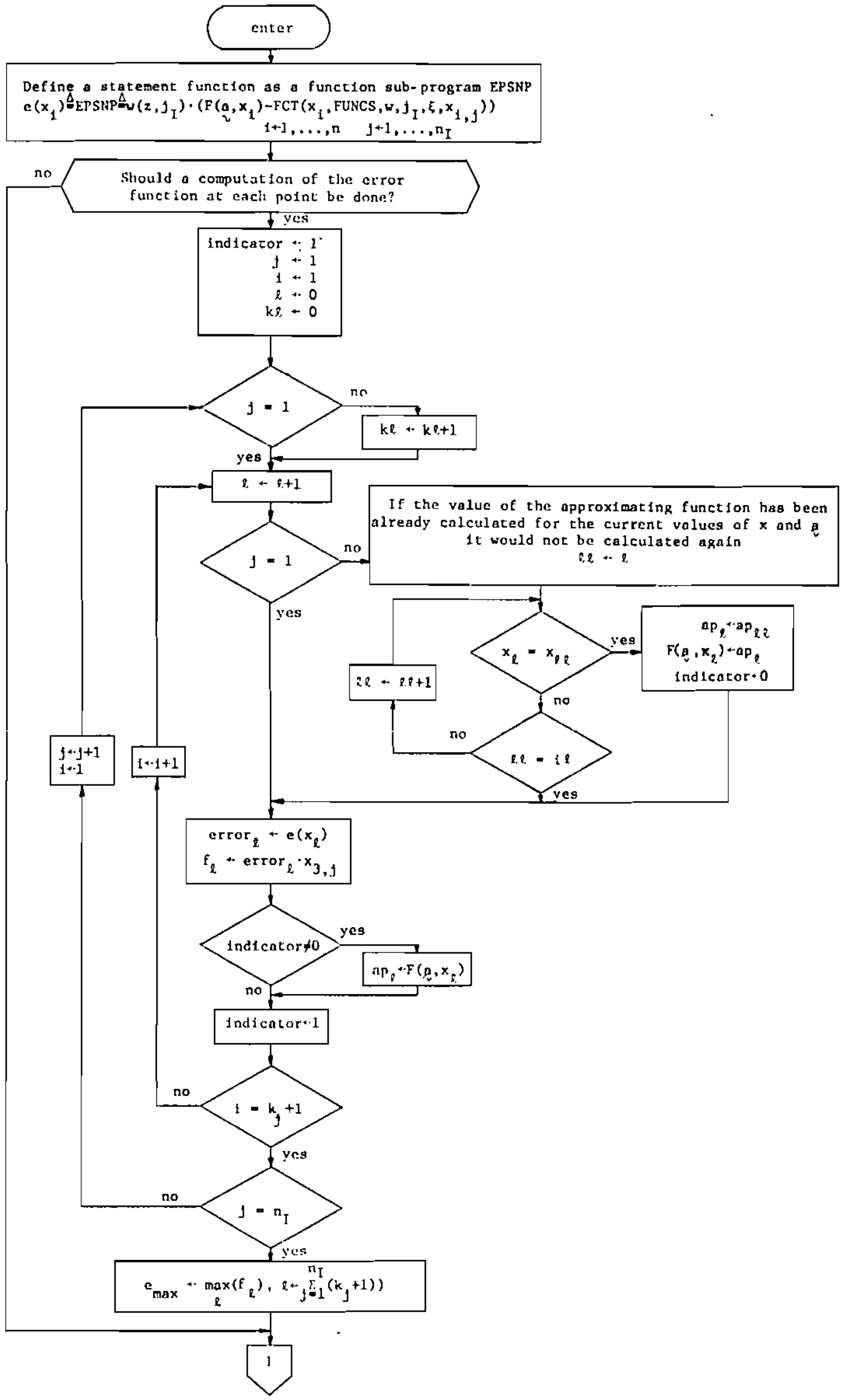
Fig. 5

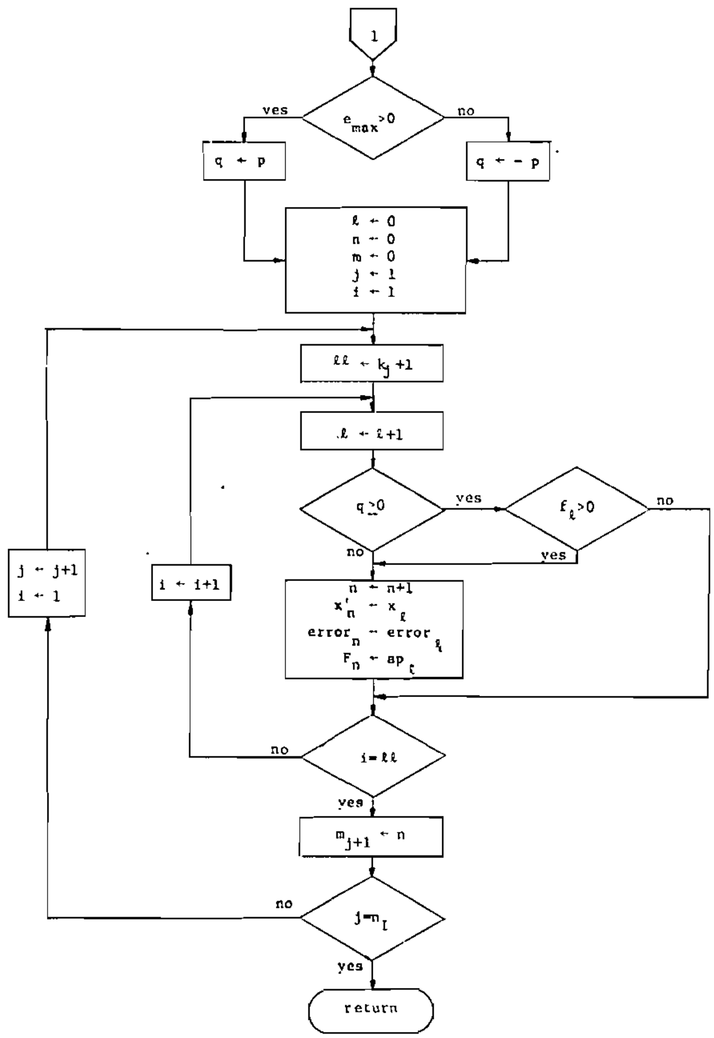

enter
Read
k_j, x_{i,j}, i←1,2,3 and j←1,...,n_I
a_i, i←1,...,k
p, ε
Discrete point set to be read?
yes
no
ℓ ← 0
j ← 1
Discrete points will be arranged equidistantly in each interval
j ← 1
ℓ ← ℓ+1
kℓ ← ℓ+k_j
j←j+1
x_i ← x_{1,j}+(x_{2,j}−x_{1,j})·(i−1)/k_j, i←1,...,k_j+1
Read
x_i, i←ℓ,...,kℓ
j ← j+1
no
j=n_I
yes
no
j=n_I
yes
Total number of discrete points is
n←Σ_{j=1}^{n_I}(k_j+1)
Calculate
error_i ≜ 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
1
STOP



Flowchart of subroutine FMLPO.

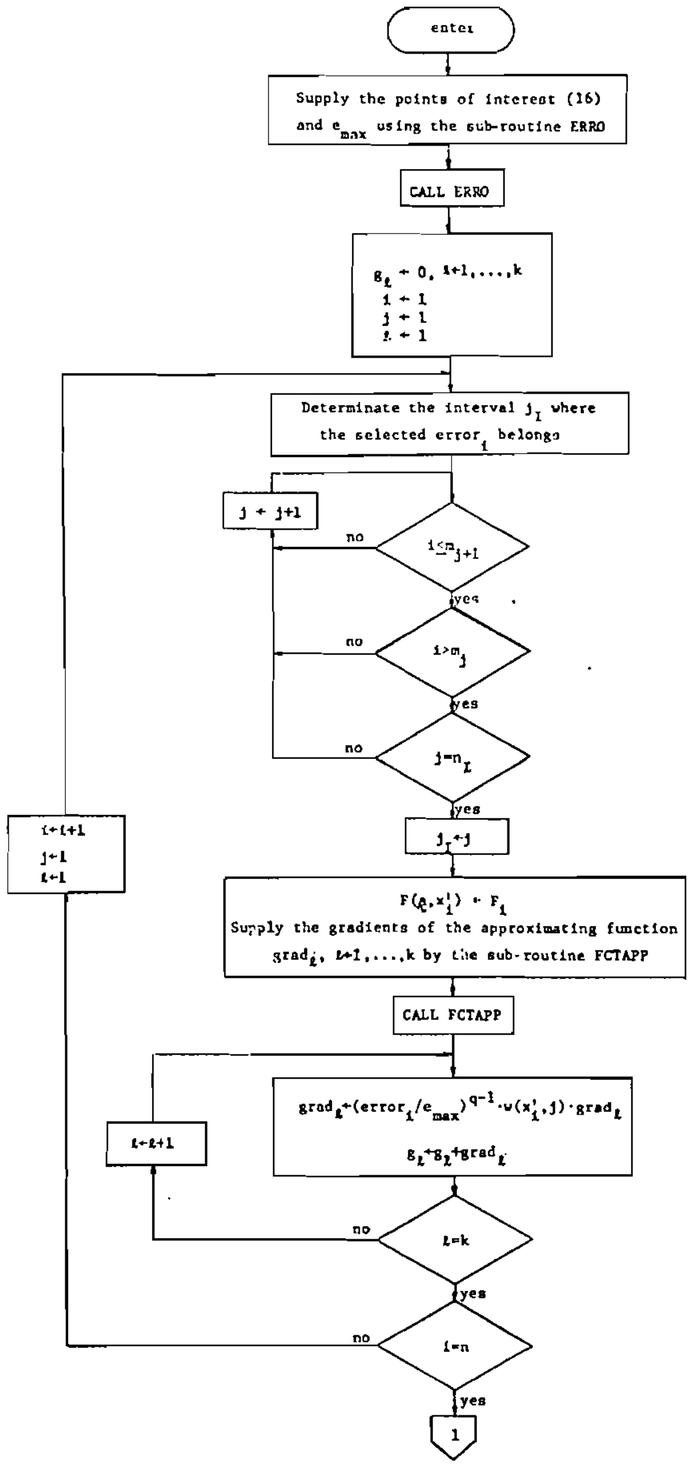
Fig. 6

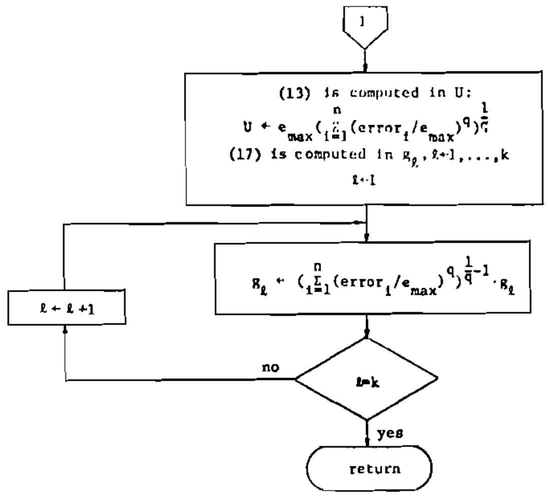

enter
Define a statement function as a function sub-program EPSNP
i←1,...,n j←1,...,n_I
no
Should a computation of the error function at each point be done?
yes
indicator ← 1
j ← 1
i ← 1
ℓ ← 0
kℓ ← 0
j = 1
no
kℓ ← kℓ+1
yes
ℓ ← ℓ+1
j = 1
no
If the value of the approximating function has been already calculated for the current values of x and a it would not be calculated again
ℓℓ ← ℓ
yes
x_ℓ = x_ℓℓ
yes
indicator←0
ℓℓ ← ℓℓ+1
no
no
ℓℓ = iℓ
yes
j←j+1
i←1
i←i+1
indicator≠0
yes
no
indicator←1
no
yes
no
yes



Flowchart of subroutine ERRO.

Fig. 7


enter
Supply the points of interest (16) and e max using the sub-routine ERRO
CALL ERRO
Determinate the interval j where the selected error belongs
j ← j+1
no
yes
F(a,x) ← F
Supply the gradients of the approximating function
by the sub-routine FCTAPP
CALL FCTAPP
no
yes
1



Flowchart of subroutine FUNGT.

FUNCS, W, FCT and EPSNP are function sub-programmes, and the others are subroutine sub-programmes.

A user of FMLPO writes FUNCS, FCTAPP and W. The intervals, not necessarily disjoint, are arranged 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 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, which relates the type of the specification to the appropriate intervals.

This programme terminates under the same conditions as the previous one.

7. Examples

Example 1

FMCLP is used to approximate

$$f(x)=\frac{\sqrt{[(8x-1)^2+1]}\tan^{-1}(8x)}{8x}, \tag{23}$$

with $w(x)=1$ on $[-1, 1]$ by a rational function

$$F(x)=\frac{a_0+a_1x+a_2x^2}{1+b_1x+b_2x^2}. \tag{24}$$

This example is remarkable because it works near degeneracy defined in the Chebyshev theorem (Achieser 1956), and meets the artificial poles for almost every combination of polynomials in the numerator and denominator of the rational function.

The initial approximation and the starting point set in the example were chosen to be the same as in the minimax approximation problem (Werner, Stoer and Bommas 1967). The initial approximation was obtained by rational interpolation where the zeros of the $(k+1)$th order Chebyshev polynomial T_{k+1} transformed on $[x_a, x_b]$ are used as supporting points. As a first trial for the working set the extrema of the $(n+1)$th order Chebyshev polynomial were taken. There is no special reason for choosing these initial points with the least pth objective, but it is shown that they provide a good initial guess for the minimax algorithm.

The comparison between the minimax and least pth approximation obtained by the Fletcher and Fletcher–Powell optimization techniques for $p=10^4$ are presented in tables 1 and 2.

We conclude that good results were obtained with both methods. The Fletcher method was more efficient than the Fletcher–Powell method.

Example 2

The second example is another that might be expected to give trouble. Due to Curtis and Powell (1965), it is the approximation of x^2 by $a_1x+a_2e^x$ over $0 \leqslant x \leqslant 2$. It may be verified that the error function of the approximation

$$x^2 \approx 8{\cdot}465x - 2{\cdot}0239e^x \tag{25}$$

takes its maximum absolute value at $x=0$, $x=1{\cdot}1227$ and $x=2$, the error at these points being $+2{\cdot}0239$, $-2{\cdot}0239$ and $+2{\cdot}0239$, respectively. In fact, the best approximation is

$$x^2 \approx 0{\cdot}1842x + 0{\cdot}4186e^x, \tag{26}$$

the maximum absolute error is $0{\cdot}5382$ and this error occurs at just the two points: $x=0{\cdot}4064$ and $x=2$. Not only do the approximating functions fail to form a Chebyshev set (Ralston 1965), but also the error curve has only two extrema instead of the three that would normally be anticipated according to the Rice's theorem (Rice 1960). The least pth results for this problem are given in table 3 and again show the success of FMCLP. The estimates of the best approximation agree to four figures with those given by Curtis and Powell.

Example 3

FMCLP is used to find a second-order model of a fourth-order system with a given transfer function

$$G(s) = \frac{s+4}{(s+1)(s^2+4s+8)(s+5)}. \tag{27}$$

The transfer function of the second-order model considered is

$$H(s) = \frac{a_3}{(s+a_1)^2 + {a_2}^2}. \tag{28}$$

Using the inverse Laplace transform the responses for (27) and (28) are

$$S(t) = \tfrac{3}{20}e^{-t} + \tfrac{1}{52}e^{-5t} - \tfrac{1}{65}e^{-2t}\,(3\sin 2t + 11\cos 2t) \tag{29}$$

and

$$F(\mathbf{a}, t) = \frac{a_3}{a_2}\,e^{-a_1t}\sin a_2t, \tag{30}$$

Table 1. Results for example 1

	Starting values	Minimax solution	Least *p*th solution for $p = 10^4$	
			Fletcher–Powell	Fletcher
a_0	$1 \cdot 00000 \times 10^{-2}$	$1 \cdot 41450$	$1 \cdot 41465$	$1 \cdot 41448$
a_1	$-3 \cdot 33600$	$-1 \cdot 06530 \times 10^1$	$-1 \cdot 06524 \times 10^1$	$-1 \cdot 06519 \times 10^1$
a_2	$4 \cdot 76782 \times 10^1$	$4 \cdot 16169 \times 10^1$	$4 \cdot 16156 \times 10^1$	$4 \cdot 16157 \times 10^1$
b_1	$1 \cdot 76567$	$-4 \cdot 01026$	$-4 \cdot 00994$	$-4 \cdot 00940$
b_2	$3 \cdot 19620 \times 10^1$	$2 \cdot 82628 \times 10^1$	$2 \cdot 82615 \times 10^1$	$2 \cdot 82620 \times 10^1$
U or M		$M = 2 \cdot 38113 \times 10^{-2}$	$U = 2 \cdot 37906 \times 10^{-2}$	$U = 2 \cdot 38154 \times 10^{-2}$
Number of runs		8	3	13
Number of function evaluations		—	1816	586
Execution time in seconds		—	43	9·2

Table 2. Results for example 1

Minimax solution		Least pth solution for $p=10^4$			
		Fletcher–Powell		Fletcher	
x where the extreme errors occur	Extreme errors	Working set of x at the optimum	Errors at the optimum	Working set of x at the optimum	Errors at the optimum
-1.00000	-2.38108×10^{-2}	-1.00000	-2.37877×10^{-2}	-1.00000	-2.38025×10^{-2}
		-9.59493×10^{-1}	-2.15681×10^{-2}	-9.59493×10^{-1}	-2.15823×10^{-2}
		-7.27181×10^{-1}	-6.38268×10^{-3}	-4.34766×10^{-1}	1.77356×10^{-2}
-3.08573×10^{-1}	2.38108×10^{-2}	-6.54861×10^{-1}	-7.35579×10^{-4}	-3.15535×10^{-1}	2.38026×10^{-2}
-6.15510×10^{-2}	-2.38109×10^{-2}	-3.16085×10^{-1}	2.37878×10^{-2}	-6.18947×10^{-2}	-2.38007×10^{-2}
5.42007×10^{-2}	2.38108×10^{-2}	-6.38133×10^{-2}	-2.37859×10^{-2}	5.46342×10^{-2}	2.37994×10^{-2}
		5.31938×10^{-2}	2.37845×10^{-2}	1.93965×10^{-1}	-2.37976×10^{-2}
1.96670×10^{-1}	-2.38113×10^{-2}	1.97478×10^{-1}	-2.37842×10^{-2}	1.94011×10^{-1}	-2.37976×10^{-2}
5.52892×10^{-1}	2.38108×10^{-2}	5.63157×10^{-1}	2.37857×10^{-2}	5.52292×10^{-1}	2.38004×10^{-2}
		8.41252×10^{-1}	1.25254×10^{-2}	6.20236×10^{-1}	2.27473×10^{-2}
		9.59492×10^{-1}	5.94070×10^{-3}	8.09651×10^{-1}	1.42303×10^{-2}
		1.00000	3.71747×10^{-3}	1.00000	3.68693×10^{-3}

Table 3. Results for example 2

Fletcher method: Initial approximation $[1\ 1]^T$; Stopping criteria 10^{-6}	
n	10
p	10^5
a_1	0·1848
a_2	0·4184
$M(\mathbf{a})$	0·5382
x where $M(\mathbf{a})$ occur	0·4066
	2
$U(\mathbf{a})$	0·5382
Function evaluations	67
Execution time in seconds	1·3

respectively, where

$$\mathbf{a}=\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}. \tag{31}$$

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 $0\leqslant t\leqslant 10$ are given in table 4, agreeing with those given by Bandler and Charalambous (1973).

Using 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 (18) in comparison with the other two cases under consideration.

The optimum result for a_2 is true for both positive and negative values from table 4 because this does not affect the approximating function (30) since

$$\frac{\sin a_2 t}{a_2}=\frac{\sin(-a_2 t)}{-a_2}. \tag{32}$$

FMLPO was used to solve this problem for different values of ξ, and the results agree with those given in table 4 when no quadratic interpolation was used.

Example 4

FMLPO was used in the optimization of a five-section cascaded transmission-line low pass filter which has been considered by Carlin (1971). 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_{0i}, such that a maximum insertion loss in the passband, from 0 to 1 GHz, is not more than 0·4 dB, while maximizing it at a point in the stopband. All section lengths

Table 4. Results for example 3

	$n=10$, $n_s=5$	$n=25$, $n_s=2$	$n=50$, $n_s=1$
$p=2$	$a_1=1{\cdot}27339$ $a_2=6{\cdot}54190\times10^{-1}$ $a_3=2{\cdot}17787\times10^{-1}$ $M(\mathbf{a})=2{\cdot}0586\times10^{-2}$ $t_M=2{\cdot}51460\times10^{-1}$ $U(\mathbf{a})=4{\cdot}7153\times10^{-3}$ f.e. = 41 q.i. = 2 1·6 sec	$a_1=1{\cdot}05489$ $a_2=-7{\cdot}67814\times10^{-1}$ $a_3=1{\cdot}61819\times10^{-1}$ $M(\mathbf{a})=1{\cdot}3271\times10^{-2}$ $t_M=2{\cdot}46234\times10^{-1}$ $U(\mathbf{a})=1{\cdot}2166\times10^{-2}$ f.e. = 30 q.i. = 3 2·3 sec	$a_1=1{\cdot}01687$ $a_2=7{\cdot}89151\times10^{-1}$ $a_3=1{\cdot}61435\times10^{-1}$ $M(\mathbf{a})=1{\cdot}2870\times10^{-2}$ $t_M=2{\cdot}04081\times10^{-1}$ $U(\mathbf{a})=2{\cdot}0668\times10^{-2}$ f.e. = 36 q.i. = 0 5·1 sec
$p=10$	$a_1=7{\cdot}37873\times10^{-1}$ $a_2=9{\cdot}26224\times10^{-1}$ $a_3=1{\cdot}28626\times10^{-1}$ $M(\mathbf{a})=8{\cdot}9565\times10^{-3}$ $t_M=1{\cdot}67727\times10^{-1}$ $U(\mathbf{a})=8{\cdot}4364\times10^{-3}$ f.e. = 38 q.i. = 3 1·5 sec	$a_1=7{\cdot}46289\times10^{-1}$ $a_2=-9{\cdot}23825\times10^{-1}$ $a_3=1{\cdot}27596\times10^{-1}$ $M(\mathbf{a})=8{\cdot}7615\times10^{-3}$ $t_M=1{\cdot}73062\times10^{-1}$ $U(\mathbf{a})=8{\cdot}2421\times10^{-3}$ f.e. = 32 q.i. = 3 2·5 sec	$a_1=7{\cdot}43325\times10^{-1}$ $a_2=9{\cdot}29377\times10^{-1}$ $a_3=1{\cdot}28119\times10^{-1}$ $M(\mathbf{a})=8{\cdot}5446\times10^{-3}$ $t_M=2{\cdot}04081\times10^{-1}$ $U(\mathbf{a})=9{\cdot}1834\times10^{-3}$ f.e. = 31 q.i. = 0 4·5 sec
$p=10^2$	$a_1=6{\cdot}79369\times10^{-1}$ $a_2=9{\cdot}55429\times10^{-1}$ $a_3=1{\cdot}21997\times10^{-1}$ $M(\mathbf{a})=8{\cdot}9563\times10^{-3}$ $t_M=1{\cdot}67727\times10^{-1}$ $U(\mathbf{a})=8{\cdot}2055\times10^{-3}$ f.e. = 35 q.i. = 1 1·3 sec	$a_1=6{\cdot}80476\times10^{-1}$ $a_2=-9{\cdot}54715\times10^{-1}$ $a_3=1{\cdot}22068\times10^{-1}$ $M(\mathbf{a})=8{\cdot}7300\times10^{-3}$ $t_M=1{\cdot}73062\times10^{-1}$ $U(\mathbf{a})=8{\cdot}1866\times10^{-3}$ f.e. = 35 q.i. = 1 2·5 sec	$a_1=6{\cdot}88905\times10^{-1}$ $a_2=9{\cdot}52106\times10^{-1}$ $a_3=1{\cdot}23339\times10^{-1}$ $M(\mathbf{a})=7{\cdot}9450\times10^{-3}$ $t_M=2{\cdot}04081\times10^{-1}$ $U(\mathbf{a})=8{\cdot}0045\times10^{-3}$ f.e. = 35 q.i. = 0 5 sec
$p=10^3$	$a_1=6{\cdot}73700\times10^{-1}$ $a_2=9{\cdot}55909\times10^{-1}$ $a_3=1{\cdot}21680\times10^{-1}$ $M(\mathbf{a})=8{\cdot}1125\times10^{-3}$ $t_M=1{\cdot}67727\times10^{-1}$ $U(\mathbf{a})=8{\cdot}1182\times10^{-3}$ f.e. = 28 q.i. = 0 1 sec	$a_1=6{\cdot}77142\times10^{-1}$ $a_2=-9{\cdot}55568\times10^{-1}$ $a_3=1{\cdot}21735\times10^{-1}$ $M(\mathbf{a})=8{\cdot}0905\times10^{-3}$ $t_M=1{\cdot}73062\times10^{-1}$ $U(\mathbf{a})=8{\cdot}0957\times10^{-3}$ f.e. = 29 q.i. = 0 2·3 sec	$a_1=6{\cdot}85101\times10^{-1}$ $a_2=9{\cdot}52890\times10^{-1}$ $a_3=1{\cdot}22946\times10^{-1}$ $M(\mathbf{a})=7{\cdot}9009\times10^{-3}$ $t_M=2{\cdot}04082\times10^{-1}$ $U(\mathbf{a})=7{\cdot}9068\times10^{-3}$ 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

f.e. denotes the number of function evaluations.
q.i. denotes the number of quadratic interpolations.

were kept fixed at 2·5 cm so that the maximum stopband insertion loss would occur at 3 GHz and the normalized characteristic impedances are used as variables. Twenty-one uniformly spaced sample points were used in the passband and a single point at 3 GHz. 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 **a** was as given by Carlin (1971):

$$[3{\cdot}180 \quad 0{\cdot}443 \quad 4{\cdot}38 \quad 0{\cdot}443 \quad 3{\cdot}180]^T. \tag{33}$$

Results obtained using the Fletcher method for $p=10^3$ are presented in table 5 and the response is shown in fig. 8.

Table 5. Results for example 4—Fletcher method

$p=10^3$	$\xi=0$	$\xi=2{\cdot}337\times10^{-2}$
a_1	3·1525	3·1508
a_2	$4{\cdot}4203\times10^{-1}$	$4{\cdot}4165\times10^{-1}$
a_3	4·4212	4·4194
a_4	$4{\cdot}4159\times10^{-1}$	$4{\cdot}4169\times10^{-1}$
a_5	3·1526	3·1508
M	$3{\cdot}9466\times10^{-5}$	$-2{\cdot}3330\times10^{-2}$
x_M	$3{\cdot}0700\times10^{-1}$	3·0
U	$3{\cdot}9466\times10^{-5}$	$-2{\cdot}3330\times10^{-2}$
Function evaluations	177	79
Execution time in seconds	17	13

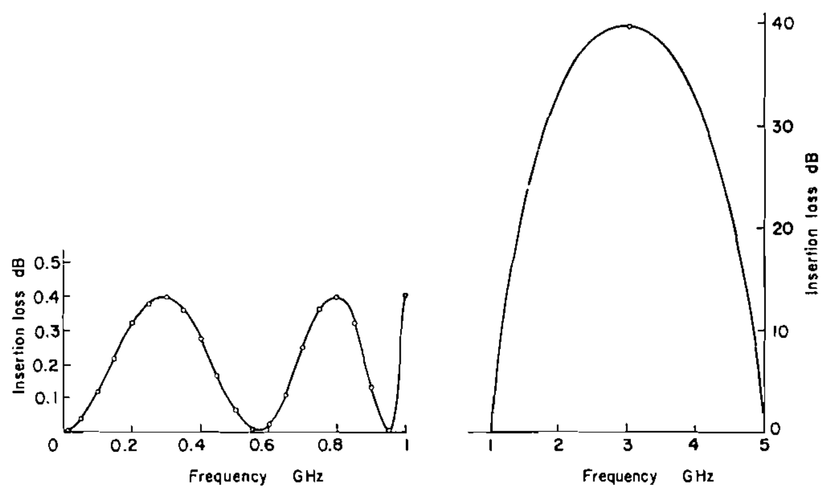


Figure 8. Response of the five section transmission-lime filter for the unconstrained design problem.

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

Example 5

Constraints are put on the parameter vector **a** such that they are not satisfied at the optimal solution of the unconstrained problem given in Example 4.

Although FMLPO is not written for non-linear programming, the constrained problem may also 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. The constraints are

$$0{\cdot}2 \leqslant a_i \leqslant 4{\cdot}0, \quad i=1, 2, \ldots, 5. \tag{34}$$

For the problem given in Example 4 and considering (34), results are presented in table 6. The response is shown in fig. 9.

Table 6. Results for example 5—Fletcher method

$p=10^3$	$\xi=0$	$\xi=2{\cdot}337\times10^{-2}$
a_1	2·9429	2·9422
a_2	$4{\cdot}1069\times10^{-1}$	$4{\cdot}1070\times10^{-1}$
a_3	4·0	3·9998
a_4	$4{\cdot}1069\times10^{-1}$	$4{\cdot}1070\times10^{-1}$
a_5	2·9429	2·9422
M	$4{\cdot}7403\times10^{-5}$	$-2{\cdot}3322\times10^{-2}$
x_M	$8{\cdot}02\times10^{-1}$	3·0
U	$4{\cdot}7403\times10^{-5}$	$-2{\cdot}3322\times10^{-2}$
Function evaluations	55	157
Execution time in seconds	31	55

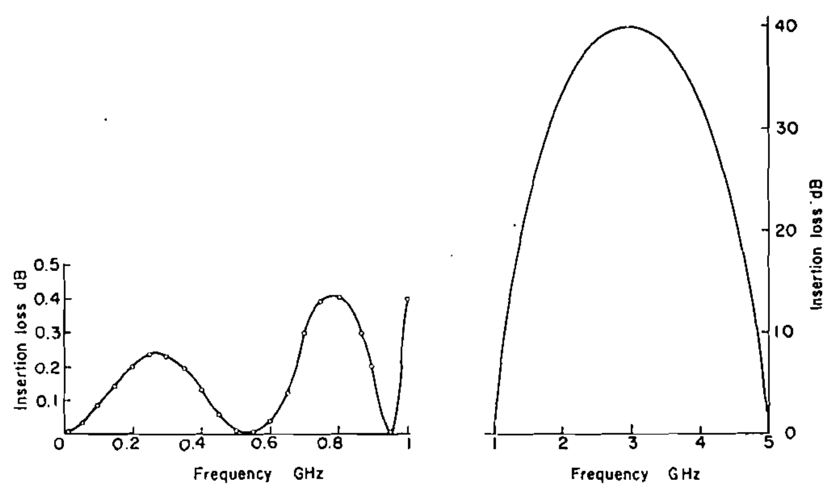


Figure 9. Response of the five section transmission-line filter for the constrained design problem.

8. Conclusions

The methods presented abandon the linear programming sub-problem which many of the minimax methods use. The advantage over the direct minimax methods is that they use very efficient gradient methods such as

Fletcher–Powell and Fletcher. 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 a minute of CDC 6400 computer time is sufficient to optimize the type of examples given in this paper to a high degree of accuracy.

The computer programme can be rearranged such that a single specified function and the approximating function are both complex (Bandler and Charalambous 1971) which is useful in electrical engineering design problems.

Information concerning the availability of the programmes can be obtained from the second author.

Acknowledgments

This work was supported by grants A7239 and C154 from the National Research Council of Canada.

Appendix

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 sub-interval ;

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

indic(2) may have values 1 or 2 :

indic(2) = 1 indicates that quadratic interpolation was not applied on a previous sub-interval ;

indic(2) = 2 indicates that a left end point of the sub-interval was temporarily removed and if indic(1) = 2, the left end point will be set to the new extreme point on the sub-interval, otherwise it will be fixed to the value it has 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 $[x_a, x_b]$.

The list of the symbols used in the flowcharts corresponding to the programme FMLPO :

error_i	corresponds to $e_{ui}'(\mathbf{a}, \xi)$ or $e_{li}'(\mathbf{a}, \xi)$,
$e_{\max}$	corresponds to $M(\mathbf{a}, \xi)$,
n_{I}	is a total number of intervals,
j_{I}	is a current interval,
k_j	is a number of subintervals of (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,
ap_i	is a value of the approximating function $F(\mathbf{a}, x_i)$,
x_i'	is an independent variable belonging to index set K (16)
m_j	is an integer which determines the interval where the selected point belongs,
grad_i	is the gradient of the approximating function $\nabla F(\mathbf{a}, x_i)$.

References

Achieser, N. I., 1956, *Theory of Approximation*, translated by C. J. Hyman (New York : Frederick Ungar Publ. Co.).

Bandler, J. W., and Charalambous, C., 1971, *5th Asilomar Conf. Circuits Systems*, Pacific Grove, California ; 1972, *I.E.E.E. Trans. Microw. Theory Tech.*, **20,** 834 ; 1973, *J. Optimization Theory Applic.* **11,** 556.

Carlin, H. J., 1971, *Proc. Instn elect. Engrs*, **59,** 1059.

Curtis, A. R., and Powell, M. J. D., 1965, *Computer J.*, **8,** 358.

Fletcher, R., 1970, *Computer J.*, **13,** 317.

Fletcher, R., and Powell, M. J. D., 1963, *Computer J.*, **6,** 163.

Ralston, A., 1965, *Numer. Math.*, **7,** 322.

Rice, J. R., 1960, *Trans. Am. Math. Soc.*, **96,** 322.

Werner, H., Stoer, J., and Bommas, W., 1967, *Numer. Math.*, **10,** 289.