INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-71

EXTRAPOLATION IN LEAST pTH APPROXIMATION AND NONLINEAR PROGRAMMING

W.Y. Chu

December 1974

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

AVAILABILITY OF REPORTS

General Contributions of manuscripts to the collection of SOC reports are on the initiative of the authors. They are not solicited.

Catalog A catalog of SOC reports including abstracts, descriptions of contents and indication of related work is available on request with orders of $10.00 or more (payable in advance).

Charges To offset preparation, printing and distribution costs the nominal charges as indicated must be made. There is a minimum charge of $10.00 per mail order. A minimum subscription order of $100.00 will entitle the subscriber to past or future reports up to the accumulated value of his subscription. A minimum order of $500.00 will entitle the subscriber to a 50% discount, however, excessive shipment expenses are to be borne by the subscriber.

Ordering Any number of these reports may be ordered. (Special reduced rates will be quoted for multiple copies of the same report). Cheques should be made out in U.S. or Canadian dollars and made payable to "McMaster University". Requests must be addressed to:

Dr. J.W. Bandler
Coordinator, G-SOC
Faculty of Engineering
McMaster University
Hamilton, Canada L8S 4L7

Reports will not normally be sent out until payment is received. Any official requisition or order not accompanied by the correct advance payment will be returned to sender.

Restrictions Reports may at any time be restricted for internal use only. Some may be revised, withdrawn or superceded. Availability is not guaranteed and descriptions or charges are subject to change without notice. The right is reserved to substitute a similar report or reprint containing essentially the same technical material in lieu of any particular request unless written instructions to the contrary are explicity supplied.

Responsibility Neither authors of reports nor any member of the Group will accept any responsibility for the consequences of any use to which the reports themselves or their contents, in particular the computer programs, are put once they leave McMaster University. No responsibility to supply any future revisions or corrections of any kind will be entertained. No responsibility will be accepted for material lost in transit, not arriving by a specified date, or arriving in a damaged or unusable condition.

EXTRAPOLATION IN LEAST pTH APPROXIMATION

AND NONLINEAR PROGRAMMING

EXTRAPOLATION IN LEAST pTH APPROXIMATION AND NONLINEAR PROGRAMMING

by

Wing Y. Chu, B. Eng.

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Engineering

McMaster University

December 1974

MASTER OF ENGINEERING (1974)
(Electrical Engineering)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE : Extrapolation in Least pth Approximation and Nonlinear Programming

AUTHOR : Wing Y. Chu
B. Eng. (McMaster University)

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

NUMBER OF PAGES: vi, 69

SCOPE AND CONTENTS:

Theoretical considerations and computational merits of applying an extrapolation technique in solving minimax problems and nonlinear programming problems using a sequence of least pth approximations or sequential unconstrained minimization techniques is presented. Numerical results indicate that the new least pth approach using extrapolation is competitive with other established minimax algorithms. An efficient, user-oriented computer program, called FLNLP2, incorporating the extrapolation technique and other recent optimization techniques is also developed. The program is capable of solving constrained or unconstrained general optimization problems and is readily applicable to circuit design problems. The extrapolation technique has been illustrated in solving the Beale problem, the Rosen-Suzuki problem, an LC lowpass filter design problem and other test examples.

ACKNOWLEDGEMENTS

The author is greatly indebted to Dr. J. W. Bandler and Dr. W. Kinsner for their guidance and encouragement throughout the course of this work, as well to Dr. E. Della Torre and Dr. C. Charalambous whose work also motivated this investigation.

The author wishes to acknowledge with thanks the useful discussions he had with his colleagues B. L. Bardakjian, P. C. Liu, M. R. M. Rizk and former colleagues J. H. K. Chen, now with Bell-Northern Research, Ottawa and J. R. Popović, now with Atomic Energy of Canada Limited, Mississauga.

Financial support provided by the Defence Research Board of Canada through grant 9931-39 and by the National Research Council of Canada through grants A7239 and E3556 and through the award of an NRC scholarship is gratefully acknowledged.

The competence and reliability of Mrs. H. Kennelly, who typed this thesis, has been very much appreciated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

To solve an optimization problem as efficiently as possible has always been a major goal in developing optimization techniques. In problems where the final solution vector is obtained as the final entry of a converging sequence of solution vectors, for example, solving nonlinear programming problems with sequential unconstrained minimization techniques [1], solving minimax problems with least pth approximation, the process may be slow. It is often desirable to have acceleration techniques to speed up convergence.

In solving nonlinear programming problems using sequential unconstrained minimization techniques, Fiacco and McCormick [1]-[2], and Lootsma [3] have employed an extrapolation technique on the sequence of unconstrained minima to accelerate convergence to the optimal solution. In this thesis, theoretical validation and computational merits of applying the same extrapolation technique in solving minimax and related problems using a sequence of least pth approximations are examined. An outcome is the development of an efficient, user-oriented computer program called FLNLP2, written in standard FORTRAN IV, which solves constrained or unconstrained general optimization problems. The Bandler-Charalambous minimax formulation [4], generalized least pth objective [5], the 1972 version of Fletcher's method [6] and the extrapolation technique are the main features of the program.

A review of the theoretical background and computational implications of the extrapolation technique used by Fiacco and McCormick is given in Chapter 2.

It is well known that least pth approximation with a very large value of p can, in principle, be used to achieve a near minimax solution [7]-[9]. For numerical efficiency, the process may be accomplished by using a sequence of least pth approximations with increasing values of p. In Chapter 3, several examples are used to investigate the effectiveness of the extrapolation technique in yielding accurate estimates of the minimax solution when applied to a sequence of least pth minima. Where appropriate, the present new approach is compared with other existing techniques for solving minimax problems. Numerical results indicate that the new approach is competitive and a theoretical analysis of the trajectory of least pth minima confirms the validity of the extrapolation procedure.

Problems for future investigation and applicability of the extrapolation technique to other minimax algorithms are discussed in Chapter 4. A complete FORTRAN listing of FLNLP2 together with the documentation for the user is given in Appendix A.

Most of the numerical results were obtained from the CDC 6400 computer, some from the PDP 11/45 computer. Parts of this work have been published, and appear in references [10]-[11].

CHAPTER 2

NONLINEAR PROGRAMMING AND EXTRAPOLATION

2.1 Introduction

In solving nonlinear programming problems, transformation techniques are usually employed to transform the constrained optimization problem into one or more unconstrained optimization problems. One of the well-established approaches is the sequential unconstrained minimization technique due to Fiacco and McCormick [1]. In using the sequential unconstrained minimization technique, Fiacco and McCormick [1]-[2] and Lootsma [3] have shown that, under certain assumptions, the problem variables, on the trajectory of minima of the sequence of unconstrained functions, can be developed as functions of the parameter r. This provides a theoretical basis for an extrapolation technique that significantly accelerates convergence to the optimal solution. In this chapter, a review of the theoretical background and computational implications of the extrapolation procedure will be given.

2.2 Interior-point Unconstrained Minimization Techniques

The nonlinear programming problem is defined as

$$\left.\begin{aligned} &\text{Minimize} \\ &\pounds \triangleq f(\underset{\sim}{\phi}) \\ &\text{subject to} \\ &g_i(\underset{\sim}{\phi}) \geq 0 \ , \ i = 1,2,\ldots,m, \end{aligned}\right\} \qquad (2.1)$$

where f is the objective function, the vector $\underset{\sim}{\phi}$ represents a set of n variables

$$\underset{\sim}{\phi} \triangleq [\phi_1 \ \phi_2 \ \cdots \ \phi_n]^T , \qquad (2.2)$$

and $g_1(\underset{\sim}{\phi})$, $g_2(\underset{\sim}{\phi})$, ..., $g_m(\underset{\sim}{\phi})$ are the constraint functions. Both f and the g_i's are, in general, nonlinear differentiable functions of the variables. The feasible region of the constrained problem is defined as

$$R_c \triangleq \{\underset{\sim}{\phi} | g_i(\underset{\sim}{\phi}) \geq 0 \quad , \quad i = 1,2,\ldots,m\}. \qquad (2.3)$$

The interior of the feasible region is the set

$$R_c^o \triangleq \{\underset{\sim}{\phi} | g_i(\underset{\sim}{\phi}) > 0 \quad , \quad i = 1,2,\ldots,m\}. \qquad (2.4)$$

Problem (2.1) can be reformulated as follows. Minimize

$$U(\underset{\sim}{\phi},r) = f(\underset{\sim}{\phi}) + r \sum_{i=1}^{m} G_i(g_i(\underset{\sim}{\phi})) \qquad (2.5)$$

where r is a positive controlling parameter and $G_i(t)$ is defined continuously on the interval $t > 0$ such that $G_i(t) \to \infty$ as $t \to 0^+$. With this formulation, a barrier is created at the boundary of the feasible region R_c and the minimal solution, $\check{\underset{\sim}{\phi}}$, is approached from the interior of R_c (i.e.,$R_c^o \neq \emptyset$) by modifying the barrier using the controlling parameter. For any $r > 0$, a point $\underset{\sim}{\phi}(r)$ minimizing (2.5) over R_c^o exists. Any convergent sequence $\{\underset{\sim}{\phi}(r_i)\}$, where $\{r_i\}$ is a monotonic decreasing null sequence as $i \to \infty$, converges to a solution of (2.1). The method is called an interior-point method or barrier function method.

There are three interior-point methods that have attracted considerable theoretical and computational attention. Firstly, there is the logarithmic programming method with

$$G_i(g_i(\underset{\sim}{\phi})) = -\ln(g_i(\underset{\sim}{\phi})). \tag{2.6}$$

It was originally proposed by Frisch [12], and further developed by Lootsma [3], [13]. Secondly, we find the interior-point method using an inverse barrier function, i.e.,

$$G_i(g_i(\underset{\sim}{\phi})) = (g_i(\underset{\sim}{\phi}))^{-1}. \tag{2.7}$$

It was first suggested by Carroll [14], and further developed by Fiacco and McCormick [1]. Lastly, there is the interior-point method with

$$G_i(g_i(\underset{\sim}{\phi})) = (g_i(\underset{\sim}{\phi}))^{-2} \tag{2.8}$$

as described by Kowalik [15], and Fletcher and McCann [16].

2.3 Analysis of Isolated Trajectory

We shall impose the following conditions on problem (2.1):

(C1) The functions $f, -g_1, \ldots, -g_m$ are convex and twice-differentiable.

(C2) The constraint set R_c is compact and its interior R_c^o is non-empty.

(C3) The Hessian matrix of the unconstrained objective function U defined by (2.5) is nonsingular for any $\underset{\sim}{\phi} \varepsilon R_c^o$ and for every $r > 0$.

We may note that U is convex on R_c^o by virtue of condition (C1). Clearly, condition (C3) implies the strict convexity of U on R_c^o. Hence, for every positive r, a unique point $\underset{\sim}{\phi}(r) \varepsilon R_c^o$ exists minimizing U over R_c^o. Let $\underset{\sim}{\nabla} f$ and $\underset{\sim}{\nabla} g_i$ denote the gradients of f and

g_i, i = 1, ..., m, respectively, where

$$\underset{\sim}{\nabla} = \left[\frac{\partial}{\partial \phi_1} \frac{\partial}{\partial \phi_2} \cdots \frac{\partial}{\partial \phi_n}\right]^T . \tag{2.9}$$

Observing that the gradient of U vanishes at $\underset{\sim}{\phi}(r)$, we find that, for a logarithmic barrier function (2.6), $\underset{\sim}{\phi}(r)$ solves the system of equations

$$\underset{\sim}{\nabla} f(\underset{\sim}{\phi}) - r \sum_{i=1}^{m} \frac{\underset{\sim}{\nabla} g_i(\underset{\sim}{\phi})}{g_i(\underset{\sim}{\phi})} = \underset{\sim}{0}. \tag{2.10}$$

Since by assumption the Hessian matrix of U is nonsingular, the implicit function theorem [17] assures us that $\underset{\sim}{\phi}(r)$ is a continuously differentiable vector function of r for r > 0. In other words, there exists an isolated trajectory of local unconstrained minima of U in R_c^o. It can be shown that this trajectory has an order of differentiability with respect to the parameter r one less than that of the original problem functions and that it is analytic when the functions are analytic (see Fiacco and McCormick [1]).

The main question is, however, the convergence of $\underset{\sim}{\phi}(r)$ to a minimal solution $\underset{\sim}{\check{\phi}}$ of problem (2.1) as $r \to 0$. Therefore, (2.10) has to be modified in such a way that conclusions on the behaviour of $\underset{\sim}{\phi}(r)$ can also be drawn in the limiting case where $r \to 0$. Let us consider the system

$$\left.\begin{aligned} \underset{\sim}{\nabla} f(\underset{\sim}{\phi}) - \sum_{i=1}^{m} u_i \underset{\sim}{\nabla} g_i(\underset{\sim}{\phi}) &= \underset{\sim}{0} , \\ u_i g_i(\underset{\sim}{\phi}) - r = 0, \ i &= 1,\ldots,m. \end{aligned}\right\} \tag{2.11}$$

For any positive r, a solution of (2.11) is given by $(\underset{\sim}{\phi}(r), \underset{\sim}{u}(r), r)$ where $\underset{\sim}{u}$ represents an m-dimensional vector with components

$$u_i(r) = \frac{r}{g_i(\underset{\sim}{\phi}(r))}\ , \ i = 1,\ldots,m. \tag{2.12}$$

Under the conditions (C1) and (C2) $\check{\underset{\sim}{\phi}}$ is a minimal solution of (2.1) if an only if there exist non-negative multipliers $\check{u}_1,\ldots,\check{u}_m$ such that the equations

$$\left.\begin{aligned} \nabla f(\underset{\sim}{\phi}) - \sum_{i=1}^{m} u_i \nabla g_i(\underset{\sim}{\phi}) &= \underset{\sim}{0}, \\ u_i g_i(\underset{\sim}{\phi}) = 0\ , \ i &= 1,\ldots,m, \end{aligned}\right\} \tag{2.13}$$

are satisfied for $\underset{\sim}{\phi} = \check{\underset{\sim}{\phi}}$ and $\underset{\sim}{u} = \check{\underset{\sim}{u}}$. These are the Kuhn-Tucker relations. Taking $\check{\underset{\sim}{u}}$ to denote the m-dimensional vector with components $\check{u}_1,\ldots,\check{u}_m$, one can readily verify that $(\check{\underset{\sim}{\phi}},\check{\underset{\sim}{u}},0)$ solves (2.11). Let $\check{J}$ denote the Jacobian matrix of (2.13), evaluated at $(\check{\underset{\sim}{\phi}},\check{\underset{\sim}{u}})$. If $\check{J}$ is nonsingular, then there exists a neighbourhood of $(\check{\underset{\sim}{\phi}},\check{\underset{\sim}{u}})$, where $(\check{\underset{\sim}{\phi}},\check{\underset{\sim}{u}})$ is the unique solution of (2.13). A set of sufficient conditions for $\check{J}$ to be nonsingular are:

(C4) The gradients $\nabla g_i(\check{\underset{\sim}{\phi}})$, $i \in B$, are linearly independent, where

$$B \triangleq \{i \mid g_i(\check{\underset{\sim}{\phi}}) = 0\}. \tag{2.14}$$

(C5) The multipliers $\check{u}_i$, $i \in B$, are positive.

(C6) The matrix $H(\check{\underset{\sim}{\phi}},\check{\underset{\sim}{u}})$ is positive definite, where

$$H(\underset{\sim}{\phi},\underset{\sim}{u}) \triangleq \underset{\sim}{\nabla}(\underset{\sim}{\nabla} f(\underset{\sim}{\phi}))^T - \sum_{i=1}^{m} u_i \underset{\sim}{\nabla}(\underset{\sim}{\nabla} g_i(\underset{\sim}{\phi}))^T. \tag{2.15}$$

The behaviour of the vector functions $(\underset{\sim}{\phi}(r),\ \underset{\sim}{u}(r))$ in the neighbourhood of $r = 0$ may now be established in the following theorem (see Fiacco and McCormick [1], and Lootsma [18]).

Theorem 1

If the functions f, $g_1,\ldots,g_m$ have continuous (k+1)th order ($k \geq 1$) partial derivatives with respect to $\underset{\sim}{\phi}$, then under conditions (C1) to (C6), the functions $(\underset{\sim}{\phi}(r), \underset{\sim}{u}(r))$ are unique and have continuous kth order derivatives in a neighbourhood about $r = 0$.

A consequence of Theorem 1 is that both $\underset{\sim}{\phi}(r)$ and $\underset{\sim}{u}(r)$ can be expanded in a Taylor series about $r = 0$. This provides a basis for extrapolation towards $(\check{\underset{\sim}{\phi}},\check{\underset{\sim}{u}})$.

Although the analysis so far is confined to interior-point methods, it can readily be extended to exterior-point methods, or mixed interior-point-exterior-point methods (see Fiacco and McCormick [1]).

2.4 Acceleration of Convergence by Extrapolation

2.4.1 Extrapolation Polynomials [1]

Suppose the unconstrained objective function $U(\underset{\sim}{\phi},r)$ has been uniquely minimized for $r_1 > \ldots > r_k > 0$ at $\underset{\sim}{\phi}(r_1),\ldots,\underset{\sim}{\phi}(r_k)$. A polynomial in r that yields $\underset{\sim}{\phi}(r_1),\ldots,\underset{\sim}{\phi}(r_k)$ is given by a set of equations of the form

$$\underset{\sim}{\phi}(r_i) = \sum_{j=0}^{k-1} \underset{\sim}{a}_j (r_i)^j \quad , \quad i = 1,\ldots,k, \qquad (2.16)$$

where the $\underset{\sim}{a}_j$ are n-component vectors. The determinant of the matrix

$$R = \begin{bmatrix} r_1^0 & \cdots & r_k^0 \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ r_1^{k-1} & \cdots & r_k^{k-1} \end{bmatrix} \tag{2.17}$$

is the Vandermonde determinant and is nonzero if $r_i \neq r_j$, $i \neq j$. Thus, there exists a unique solution for the $\underset{\sim}{a}_j$. Then $\sum_{j=0}^{k-1} \underset{\sim}{a}_j (r)^j$ is an approximation of $\underset{\sim}{\phi}(r)$ in the interval $[0,r_1]$, and $\underset{\sim}{\phi}(0) = \check{\underset{\sim}{\phi}}$ (the minimal solution) is approximated by $\underset{\sim}{a}_0$.

Now, the exact Taylor series expansion of $\underset{\sim}{\phi}(r_i)$ in r_i about $\underset{\sim}{\phi}(0)$ is

$$\underset{\sim}{\phi}(r_i) = \sum_{j=0}^{k-1} (r_i)^j \frac{D^j \underset{\sim}{\phi}(0)}{j!} + \underset{\sim}{\varepsilon}^i \quad , \quad i = 1,\ldots,k, \tag{2.18}$$

where

$$D\underset{\sim}{\phi}(r) \triangleq \left[\frac{d\phi_1(r)}{dr} \cdots \frac{d\phi_n(r)}{dr}\right]^T \quad , \tag{2.19}$$

$$\underset{\sim}{\varepsilon}^i = \left(\frac{(r_i)^k}{i!}\right) \left[\frac{d^k\phi_1(\eta_{1i})}{dr^k} \cdots \frac{d^k\phi_n(\eta_{ni})}{dr^k}\right]^T \quad , \tag{2.20}$$

$$0 \leq \eta_{ji} \leq r_i \;, \; j = 1,\ldots,n.$$

Setting (2.16) and (2.18) equal, subtracting and combining yields

$$[\underset{\sim}{\varepsilon}^1 \cdots \underset{\sim}{\varepsilon}^k] R^{-1} = A - \left[\underset{\sim}{\phi}(0) \quad \frac{D^1 \underset{\sim}{\phi}(0)}{1!} \quad \cdots \quad \frac{D^{k-1}\underset{\sim}{\phi}(0)}{(k-1)!}\right] \tag{2.21}$$

where

$$A = [\underset{\sim}{a}_0 \cdots \underset{\sim}{a}_{k-1}] . \tag{2.22}$$

Clearly, then, the difference between $\underset{\sim}{a}_0$ and $\underset{\sim}{\phi}(0)$ is of the order of r_1^k. Thus, as $r_1 \to 0$, $\underset{\sim}{a}_0 \to \underset{\sim}{\phi}(0)$. In addition, the estimates using k minima are better than those using (k-1) minima. With $r_{i+1} = r_i/c$, $c > 1$, the particular structure of these equations renders the use of an extrapolation procedure according to the Richardson-Romberg principle [19] to estimate $\underset{\sim}{a}_0$.

If $\underset{\sim}{\phi}^i_j$, $i = 1, \ldots, k$, $j = 1, \ldots, i-1$, signifies the jth order estimate of $\underset{\sim}{\phi}(0)$ after i minima have been obtained, with r_1 being the initial value of r, then we have

$$\left.\begin{aligned} &\underset{\sim}{\phi}^i_0 = \underset{\sim}{\phi}\left(\frac{r_1}{c^{i-1}}\right), \quad i = 1,\ldots,k, \\ \text{and} \quad &\underset{\sim}{\phi}^i_j = \frac{c^j\, \underset{\sim}{\phi}^i_{j-1} - \underset{\sim}{\phi}^{i-1}_{j-1}}{c^j - 1}, \quad \begin{array}{l} i = 2,\ldots,k, \\ j = 1,\ldots,i-1 . \end{array} \end{aligned}\right\} \tag{2.23}$$

The "best" estimate of $\underset{\sim}{\phi}(0)$, namely $\underset{\sim}{a}_0$, is given by

$$\underset{\sim}{\phi}(0) \cong \underset{\sim}{\phi}^k_{k-1} = \underset{\sim}{a}_0 . \tag{2.24}$$

The extrapolation formula (2.23) can also be used to estimate the next minimum of the objective function $U(\underset{\sim}{\phi},r)$, i.e., the (k+1)st minimum. Setting i = k+1 in (2.23) and solving for $\underset{\sim}{\phi}^{k+1}_{j-1}$, we have the following recursive relation

$$\underset{\sim}{\phi}^{k+1}_{j-1} = \frac{(c^j-1)\underset{\sim}{\phi}^{k+1}_j + \underset{\sim}{\phi}^k_{j-1}}{c^j} . \tag{2.25}$$

Noting that $\underset{\sim}{a}_0 = \underset{\sim}{\phi}^k_{k-1} = \underset{\sim}{\phi}^{k+1}_{k-1}$ from (2.24) and using the values previously obtained from (2.23), we can evaluate (2.25) for j = k-1, k-2,...,1. The last computation will give the required estimate $\underset{\sim}{\phi}^{k+1}_0$, This estimate can be used as the starting point for the (k+1)st minimization of $U(\underset{\sim}{\phi},r)$. As more minima are achieved, the estimate eventually improves. This accelerates the entire process by substantially reducing the effort required to minimize the successive U functions.

2.4.2 An Example [1]

To illustrate the extrapolation technique, the following example is considered.

Minimize

$$f(\underset{\sim}{\phi}) = \ln\phi_1 - \phi_2$$

subject to

$$\phi_1 \geq 1$$

$$\phi_1^2 + \phi_2^2 = 4 .$$

The analytical solution is

$$\phi_1 = 1 \quad , \phi_2 = \sqrt{3} \cong 1.7320505.$$

The sequential unconstrained minimization technique was used to solve the problem by defining

$$U(\underset{\sim}{\phi},r) = f(\underset{\sim}{\phi}) - r\ln(\phi_1 - 1) + (\phi_1^2 + \phi_2^2 - 4)^2/r \qquad (2.26)$$

and minimizing U with respect to $\underset{\sim}{\phi}$ for a decreasing sequence of r values. Extrapolation is used to accelerate convergence. Table 2.1 shows the results. The convergence of the estimates to the analytical solution can be seen by reading down the columns. The effectiveness of the extrapolation technique can be seen by noting that the third-order estimates using the last four minima agree with the analytical solution to seven significant figures, whereas the minimum for $r = 3.960625\times10^{-3}$ agrees to only three significant figures.

r_i	Estimates				Estimates			
	ϕ_1^i	1st	2nd	3rd	ϕ_2^i	1st	2nd	3rd
1.0	1.5527821				1.3328309			
0.25	1.1593476				1.6413384			
		1.0282028				1.7441742		
6.25×10^{-2}	1.0398244				1.7111098			
		0.9999833				1.7343670		
			0.9981020				1.7337131	
1.5625×10^{-2}	1.0099208				1.7269401			
		0.9999529				1.7322168		
			0.9999509				1.7320735	
				0.9999802				1.7320475
3.960625×10^{-3}	1.0024774				1.7307811			
		0.9999963				1.7320614		
			0.9999992				1.7320511	
				1.0000000				1.7320507
Analytical solution		1.0				$\sqrt{3} \cong 1.7320505$		

Table 2.1 Use of extrapolation to accelerate convergence.

CHAPTER 3

EXTRAPOLATION IN LEAST pTH APPROXIMATION

3.1 Introduction

In least pth approximation, large values of p are usually required for the least pth optimal solution to be close to the optimal minimax solution [7]-[9]. Depending on how close the starting point is to the minimax optimum, the process may be unnecessarily time-consuming. To start with a small value of p and then sequentially increase it may somewhat alleviate this problem. By this approach, a sequence of least pth minima will be obtained. Under appropriate assumptions, we may expect the sequence of least pth minima to form a unique trajectory of local minima converging to the minimax optimum, and the extrapolation technique discussed in Chapter 2 can be applied to accelerate convergence. In this chapter, several test problems are used to investigate the applicability of the extrapolation technique in solving minimax problems with a sequence of least pth approximations. Numerical results indicate that the technique is successful and efficient. The credibility of the extrapolation technique is further confirmed by theoretical considerations.

3.2 Basic Formulas

3.2.1 Generalized Least pth Objective

The generalized least pth objective function [5] to be minimized with respect to $\underset{\sim}{\phi}$ is

$$U(\underset{\sim}{\phi},p) = \begin{cases} M(\underset{\sim}{\phi}) \left(\sum\limits_{i\varepsilon K} \left(\frac{e_i(\underset{\sim}{\phi})}{M(\underset{\sim}{\phi})} \right)^q \right)^{\frac{1}{q}} & \text{for } M(\underset{\sim}{\phi}) \neq 0 \\ 0 & \text{for } M(\underset{\sim}{\phi}) = 0 \end{cases} \quad (3.1)$$

where

$e_i(\underset{\sim}{\phi})$ is a set of m+1 real error functions (linear or nonlinear)

$\underset{\sim}{\phi} \triangleq [\phi_1 \phi_2 \ldots \phi_n]^T$, a n-component parameter vector (3.2)

$q \triangleq p \text{ sgn } M(\underset{\sim}{\phi}) \quad , \quad p > 1$ (3.3)

$M(\underset{\sim}{\phi}) \triangleq \max\limits_{i\varepsilon I} \; e_i(\underset{\sim}{\phi})$ (3.4)

$$K = \begin{cases} J \triangleq \{i | e_i(\underset{\sim}{\phi}) > 0 \; , \; i\varepsilon I\} & \text{if } M(\underset{\sim}{\phi}) > 0 \\ I \triangleq \{1,2,\ldots,m+1\} & \text{if } M(\underset{\sim}{\phi}) < 0 \; . \end{cases} \quad (3.5)$$

The gradient vector of the objective function is given by

$$\underset{\sim}{\nabla} U(\underset{\sim}{\phi},p) = \left(\sum_{i\varepsilon K} \left(\frac{e_i(\underset{\sim}{\phi})}{M(\underset{\sim}{\phi})} \right)^q \right)^{\frac{1}{q}-1} \sum_{i\varepsilon K} \left(\frac{e_i(\underset{\sim}{\phi})}{M(\underset{\sim}{\phi})} \right)^{q-1} \underset{\sim}{\nabla} e_i(\underset{\sim}{\phi}) \text{ for } M(\underset{\sim}{\phi}) \neq 0. \quad \ldots.(3.6)$$

From (3.1) and (3.6) we note that if $e_i(\underset{\sim}{\phi})$ for $i\varepsilon I$ are continuous with continuous first partial derivatives, then under the stated conditions, the objective function is continuous everywhere with continuous first partial derivatives (except possibly when $M(\underset{\sim}{\phi}) = 0$ and two or more maxima are equal).

3.2.2 Bandler-Charalambous Minimax Formulation [4]

The nonlinear programming problem defined by (2.1) is transformed into the following unconstrained objective

$$V(\underset{\sim}{\phi},\alpha) = \max_{1 \leq i \leq m} \; [f(\underset{\sim}{\phi}), \; f(\underset{\sim}{\phi}) - \alpha g_i(\underset{\sim}{\phi})] \quad (3.7)$$

where

$$\alpha > 0.$$

Sufficiently large α must be chosen to satisfy the inequality

$$\frac{1}{\alpha} \sum_{i=1}^{m} \check{u}_i < 1 \qquad (3.8)$$

where the $\check{u}_i$'s are the Kuhn-Tucker multipliers at the optimum. The minimization of $V(\underset{\sim}{\phi},\alpha)$ with respect to $\underset{\sim}{\phi}$ is a minimax problem and may be solved by minimizing the generalized least pth objective with

$$e_i(\underset{\sim}{\phi}) \triangleq f(\underset{\sim}{\phi}) - \alpha g_i(\underset{\sim}{\phi}) \ , \ i = 1,2,\ldots,m \qquad (3.9)$$

$$e_{m+1}(\underset{\sim}{\phi}) \triangleq f(\underset{\sim}{\phi}) \qquad (3.10)$$

using a very large value of p or a sequence of p values with extrapolation or one of the several recent minimax algorithms proposed by Charalambous and Bandler [20] and Charalambous [21]-[22].

3.3 Estimation of Minimax Optimum by Extrapolation

A minimax example, two test functions and an LC lowpass filter design problem were used to investigate the applicability and performance of the extrapolation formula (2.23) in estimating the minimax optimum from a sequence of least pth minima. Wherever possible, the present approach was compared with other algorithms for solving minimax problems. To facilitate the investigation, the extrapolation formula (2.23) was coded into a FORTRAN subroutine and was incorporated into two nonlinear programming packages FLNLP1 [23] and FLOPT1 [24]. The updated version of FLNLP1, called FLNLP2, is described in Appendix A. Formulas discussed in section 3.2 were used for the objective formulation. The latest version of Fletcher's method [6] was used to perform the minimization. For all examples except the LC lowpass filter problem, the minimization was terminated when the change in the parameter values on an iteration was less than 10^{-8}; 10^{-7} was used for the LC lowpass filter problem.

Example 1 : A minimax example [20]

Minimize the maximum of the following three functions,

$$e_1(\underset{\sim}{\phi}) = \phi_1^{\,4} + \phi_2^{\,2}$$

$$e_2(\underset{\sim}{\phi}) = (2 - \phi_1)^2 + (2 - \phi_2)^2$$

$$e_3(\underset{\sim}{\phi}) = 2\exp(-\phi_1 + \phi_2)$$

The minimax solution occurs at the point $\phi_1 = \phi_2 = 1$ and the maximum value is 2. The problem was formulated as a least pth approximation problem. A sequence of p values, starting with p = 4 and increasing by factors of 4 up to 1024, was used. The minimax solution is obtained by extrapolation. The results are shown in Table 3.1. The convergence of the estimates to the analytical minimax solution can be seen by reading down the columns. Table 3.1a shows the improvement the extrapolation procedure made over the basic approach in yielding the minimax solution.

Example 2: Beale constrained function [25]

Minimize

$$f(\underset{\sim}{\phi}) = 9-8\phi_1-6\phi_2-4\phi_3+2\phi_1^{\,2}+2\phi_2^{\,2}+\phi_3^{\,2}+2\phi_1\phi_2+2\phi_1\phi_3$$

subject to

p_i	Estimates ϕ_1^i	1st	2nd	3rd	Estimates ϕ_2^i	1st	2nd	3rd
4	1.0228068				0.9005678			
16	1.0109514				0.9697441			
		1.0069996				0.9928028		
64	1.0033465				0.9917309			
		1.0008115				0.9990598		
			1.0003990				0.9994769	
256	1.0008851				0.9978751			
		1.0000646				0.9999232		
			1.0000148				0.9999808	
				1.0000087				0.9999888
1024	1.0002245				0.9994649			
		1.0000043				0.9999948		
			1.0000003				0.9999996	
				1.0000001				0.9999999
Analytical solution	1				1			

Table 3.1 Results of Example 1 for starting point $\underset{\sim}{\phi}^0 = [2\ 2]^T$.

Parameters	p=4,16,64,256,1024 Order of extrapolation = 3	$p = 10^5$
ϕ_1	1.0000001	1.0000023
ϕ_2	0.9999999	0.9999945
$\max e_i(\underset{\sim}{\phi})$	2.0000000	2.0000064
Function evaluations	45	62

Table 3.1a A comparison of two approaches for solving Example 1.

$$\phi_i \geq 0 \;,\; i = 1,2,3$$

$$3-\phi_1-\phi_2-2\phi_3 \geq 0.$$

The function has a minimum $f(\check{\underset{\sim}{\phi}}) = 1/9$ at $\check{\underset{\sim}{\phi}} = [4/3\ 7/9\ 4/9]^T$. The Bandler-Charalambous technique was used to transform the constrained problem into an unconstrained minimax problem. A sequence of least pth approximations together with extrapolation was used to obtain the optimal solution. The same problem was also solved by least pth approximation with a value of p of 10^5 and by the Charalambous-Bandler algorithm with a value of p of 10. Table 3.2 gives a comparison between the three approaches. It can be seen that the extrapolation procedure outperforms the other two approaches.

Example 3: Rosen-Suzuki function [25]
Minimize

$$f(\underset{\sim}{\phi}) = \phi_1^2+\phi_2^2+2\phi_3^2+\phi_4^2-5\phi_1-5\phi_2-21\phi_3+7\phi_4$$

subject to

$$-\phi_1^2-\phi_2^2-\phi_3^2-\phi_4^2-\phi_1+\phi_2-\phi_3+\phi_4+8 \geq 0$$

$$-\phi_1^2-2\phi_2^2-\phi_3^2-2\phi_4^2+\phi_1+\phi_4+10 \geq 0$$

$$-2\phi_1^2-\phi_2^2-\phi_3^2-2\phi_1+\phi_2+\phi_4+5 \geq 0 .$$

The function has a minimum $f(\check{\underset{\sim}{\phi}}) = -44$ at $\check{\underset{\sim}{\phi}} = [0\ 1\ 2\ -1]^T$. The Bandler-Charalambous technique was used to transform the nonlinear programming problem into an unconstrained minimax problem. The minimax problem was then solved using a sequence of least pth approximations together with extrapolation; using least pth approximation

Parameters	p=4,16,64,256 α=1 Order of extrapolation =3	$p = 10^5$ α=1	p = 10 α=1
ϕ_1	1.3333333	1.3333338	1.3333335
ϕ_2	0.7777778	0.7777775	0.7777777
ϕ_3	0.4444444	0.4444437	0.4444444
$f(\underset{\sim}{\phi})$	0.1111111	0.1111114	0.1111111
$g_1(\underset{\sim}{\phi})$	1.3333333	1.3333338	1.3333335
$g_2(\underset{\sim}{\phi})$	0.7777778	0.7777775	0.7777777
$g_3(\underset{\sim}{\phi})$	0.4444444	0.4444437	0.4444444
$g_4(\underset{\sim}{\phi})$	5.07×10^{-9}	1.39×10^{-6}	8.07×10^{-9}
Function evaluations	34	78	99

Table 3.2 Results for the Beale problem for starting point $\underset{\sim}{\phi}^0 = [1\ 2\ 1]^T$.

with a value of p of 10^5; using the Charalambous-Bandler algorithm with a value of p of 10. In each case, the value of the parameter α was 10. The original constrained problem was also solved using the Fiacco-McCormick method with extrapolation. Table 3.3 compares the performance of the four approaches. We may notice that the first and the last approach (both using extrapolation) give very accurate estimates of the optimal solution. However, some of the constraints are slightly violated due to the extrapolation procedure. We note that the least pth approach with extrapolation requires the minimum number of function evaluations while the best solution is given by the Fiacco-McCormick method with extrapolation.

Example 4: An LC lowpass filter design problem

Consider the design of an LC lowpass filter as shown in Fig. 3.1; the specification requirements are that the insertion loss in the passband (ω = 0 to 1) is not to exceed 0.01 dB while the insertion loss at ω = 2.5 in the stopband is to be a maximum. Letting Γ denote the insertion loss, the design problem can be formulated as

Maximize

$$U = \Gamma(\underset{\sim}{\phi},\omega_s)$$

subject to

$$\Gamma(\underset{\sim}{\phi},\omega_i) \leq 0.01 \ , \ i = 1,\ldots,m,$$

where

ω_s is the stopband frequency point,

ω_i is a set of m sampling frequency points in the passband.

	Least pth Approach			Fiacco-McCormick Method
Parameters	p=4,12,36,108,324,972 α=10 Order of extrapolation =3	$p=10^5$ α=10	p=10 α=10	$r=1,10^{-1},10^{-2},10^{-3},10^{-4}$ Order of extrapolation =3
ϕ_1	-0.0000002	-0.0000021	-0.0000008	-0.0000000
ϕ_2	1.0000005	0.9999976	1.0000001	1.0000000
ϕ_3	1.9999999	1.9999908	2.0000009	2.0000000
ϕ_4	-1.0000002	-0.9999883	0.9999982	-1.0000000
$f(\phi)$	-44.000000	-43.999804	-43.999999	-44.000000
$g_1(\phi)$	-2.80×10^{-7}	8.56×10^{-5}	1.63×10^{-6}	-9.35×10^{-10}
$g_2(\phi)$	1.00	1.00	1.00	1.00
$g_3(\phi)$	7.57×10^{-8}	5.51×10^{-5}	-2.07×10^{-7}	-7.61×10^{-11}
Function evaluations	72	107	148	125

Table 3.3 Results for the Rosen-Suzuki problem for starting point $\phi^0 = [0\ 0\ 0\ 0]^T$.

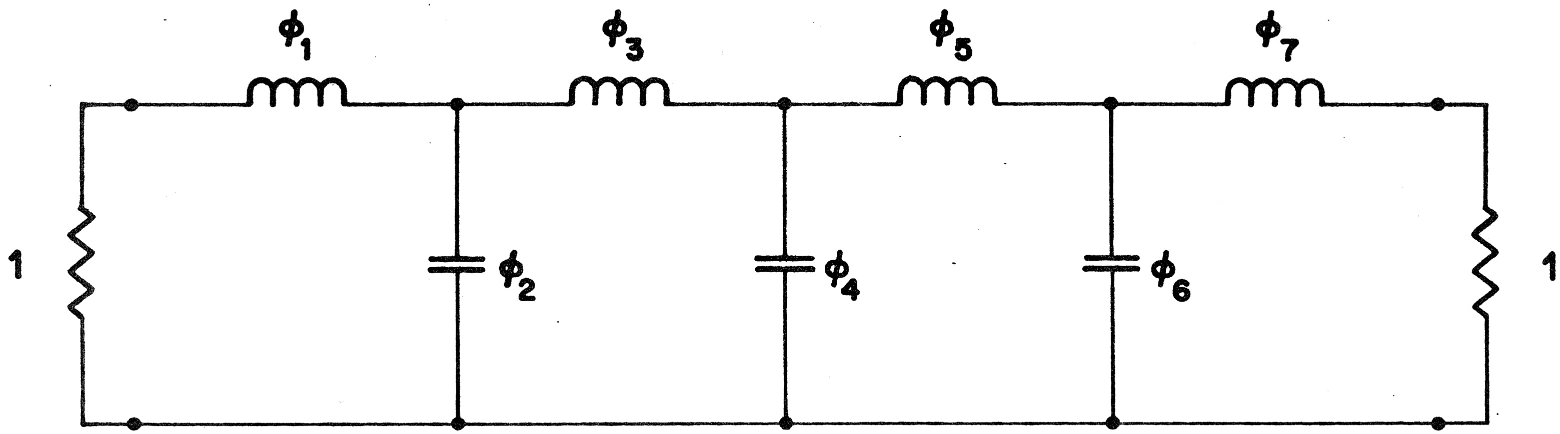

Figure 3.1 LC lowpass filter used in Example 4.

The constrained problem was transformed into an unconstrained minimax problem using the Bandler-Charalambous technique. A sequence of least pth approximations together with extrapolation was used to obtain the optimal solution. The original problem was also solved using the Fiacco-McCormick method with extrapolation. In each approach, 21 uniformly spaced sampling frequency points were used in the passband (ω = 0 to 1) and $\omega_s = 2.5$. The numerical results from a nonfeasible starting point, $\underset{\sim}{\phi}^0 = [0.7\ 1.4\ 1.5\ 1.5\ 1.5\ 1.4\ 0.7]^T$, are tabulated in Table 3.4. Discrepancies between the numerical solutions and the analytical solution are due to the finite, uniformly spaced sampling points used in the passband. Table 3.4a shows the deviation of the numerical responses from the analytical response at some crucial frequency points. We may notice that the least pth approach gives slightly better results. Fig. 3.2 shows the responses of the filter before and after optimization.

3.4 Theoretical Verification

In the previous examples, application of the extrapolation formula (2.23) is based on the assumption that the trajectory of least pth minima is a continuously differentiable function in $\frac{1}{p}$ for $1 > \frac{1}{p} \geq 0$ and can be expanded as a Taylor series about $\frac{1}{p} = 0$. It is the purpose of this section to show that, under certain conditions, our assumption is valid.

3.4.1 Assumptions

(A1) The error functions $e_i(\underset{\sim}{\phi})$ for $i \in I$ are convex and have continuous (k+1)th order, $k \geq 1$, partial derivatives with respect to $\underset{\sim}{\phi}$.

	Least pth Approach	Fiacco-McCormick Method	
Parameters	$p=4,15,64,256$ $\alpha=10^4$ Order of extrapolation =3	$r=10^{-2}, 2\times10^{-3}, \ldots, 6.4\times10^{-7}$ Order of extrapolation =3	Analytical Solution
ϕ_1	0.8000	0.8006	0.7969
ϕ_2	1.3929	1.3931	1.3924
ϕ_3	1.7502	1.7508	1.7481
ϕ_4	1.6332	1.6333	1.6331
ϕ_5	1.7502	1.7508	1.7481
ϕ_6	1.3929	1.3931	1.3924
ϕ_7	0.8000	0.8006	0.7969
$\Gamma(\underset{\sim}{\phi},\omega_i)_{max}$ dB	0.01001	0.01007	0.01000
$\Gamma(\underset{\sim}{\phi},\omega_s)$ dB	62.96	62.98	62.87
Function evaluations	138	118	-

Table 3.4 Results for the LC lowpass filter design problem.

	Least pth Approach		Fiacco-McCormick Method	
Frequency	Absolute error dB	Relative error %	Absolute error dB	Relative error %
.	.	.	.	.
.	.	.	.	.
0.20	2.66×10^{-4}	2.73	3.22×10^{-4}	3.31
0.25	2.60×10^{-4}	2.70	3.16×10^{-4}	3.29
.	.	.	.	.
.	.	.	.	.
0.60	4.42×10^{-4}	4.62	4.98×10^{-4}	5.21
0.65	4.92×10^{-4}	5.22	5.49×10^{-4}	5.83
.	.	.	.	.
.	.	.	.	.
0.90	-3.00×10^{-7}	-0.00	4.98×10^{-5}	0.50
.	.	.	.	.
1.00	9.70×10^{-6}	0.10	5.85×10^{-5}	0.59
2.50	9.14×10^{-2}	0.15	1.13×10^{-1}	0.18

Table 3.4a Response deviation of the two approaches.

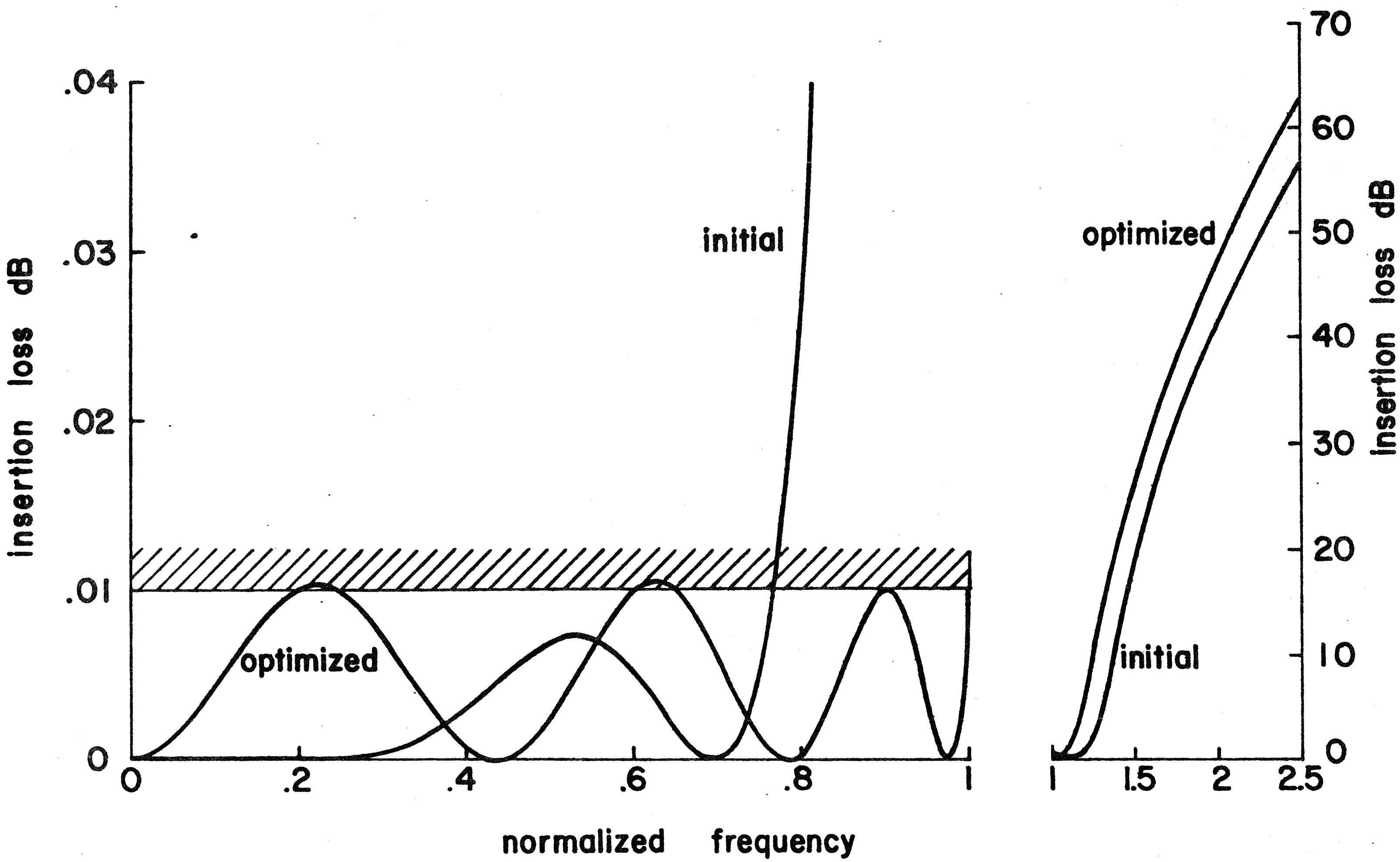

Figure 3.2 Responses of the lowpass filter.

(A2) The Hessian matrix of the objective function U is nonsingular for any real $\underset{\sim}{\phi}$ and for every $1 > \frac{1}{p} > 0$.

(A3) Assumptions (developed later) to ensure differentiability of the trajectory at $p = \infty$.

3.4.2 Analysis of Trajectory of Least pth Minima

Owing to the definiteness and convexity properties of the ojective function U, we can expect, at every value of p, an "isolated" or locally unique unconstrained minimum. Noting that the gradient of U vanishes at the minimizing point, $\underset{\sim}{\phi}(\frac{1}{p})$, we have

$$\underset{\sim}{\nabla} U(\underset{\sim}{\phi}(\tfrac{1}{p}),p) = \underset{\sim}{0} \tag{3.11}$$

or

$$\left[\sum_{i\in K}\left(\frac{e_i(\underset{\sim}{\phi}(\frac{1}{p}))}{M(\underset{\sim}{\phi}(\frac{1}{p}))}\right)^q\right]^{\frac{1}{q}-1}\sum_{i\in K}\left(\frac{e_i(\underset{\sim}{\phi}(\frac{1}{p}))}{M(\underset{\sim}{\phi}(\frac{1}{p}))}\right)^{q-1}\underset{\sim}{\nabla} e_i(\underset{\sim}{\phi}(\tfrac{1}{p})) = \underset{\sim}{0}. \tag{3.12}$$

Since by assumption the Hessian matrix of U is nonsingular, the implicit function theorem assures us that $\underset{\sim}{\phi}(\frac{1}{p})$ is a continuously differentiable vector function of $\frac{1}{p}$ for $1 > \frac{1}{p} > 0$. In other words, we have an isolated trajectory of unconstrained local minima of U.

It is possible to be explicit about the derivatives of $\underset{\sim}{\phi}(\frac{1}{p})$ with respect to $\frac{1}{p}$ for $1 > \frac{1}{p} > 0$. For convenience, let us define

$$p' \triangleq \frac{1}{p} \tag{3.13}$$

$$e_{ip'} \triangleq e_i(\underset{\sim}{\phi}(\tfrac{1}{p})) \tag{3.14}$$

$$M_{p'} \triangleq M(\underset{\sim}{\phi}(\tfrac{1}{p})). \tag{3.15}$$

Since (3.12) is an identity in $\frac{1}{q}$ (or rather p'), we can differentiate with respect to p', obtaining

$$\left[\sum_{i\varepsilon K}\left(\frac{e_{ip'}}{M_{p'}}\right)^q\right]^{\frac{1}{q}-1}\left\{\sum_{i\varepsilon K}\left(\frac{e_{ip'}}{M_{p'}}\right)^{q-1}\underset{\sim}{\nabla}(\underset{\sim}{\nabla}e_{ip'})^T D\underset{\sim}{\phi}(p')\right.$$

$$+(q-1)\left(\frac{e_{ip'}}{M_{p'}}\right)^{q-2}\left(\frac{\underset{\sim}{\nabla}e_{ip'}}{M_{p'}}\right)(\underset{\sim}{\nabla}e_{ip'})^T D\underset{\sim}{\phi}(p')$$

$$\left.-(\text{sgn } M_{p'})\, q^2\left(\frac{e_{ip'}}{M_{p'}}\right)^{q-1}\ln\left(\frac{e_{ip'}}{M_{p'}}\right)\underset{\sim}{\nabla}e_{ip'}\right\} = \underset{\sim}{0}\ . \quad (3.16)$$

Now

$$\underset{\sim}{\nabla}(\underset{\sim}{\nabla}U(\underset{\sim}{\phi}(p'),p))^T = \left[\sum_{i\varepsilon K}\left(\frac{e_{ip'}}{M_{p'}}\right)^q\right]^{\frac{1}{q}-1}\sum_{i\varepsilon K}\left\{\left(\frac{e_{ip'}}{M_{p'}}\right)^{q-1}\underset{\sim}{\nabla}(\underset{\sim}{\nabla}e_{ip'})^T\right.$$

$$\left.+(q-1)\left(\frac{e_{ip'}}{M_{p'}}\right)^{q-2}\left(\frac{\underset{\sim}{\nabla}e_{ip'}}{M_{p'}}\right)(\underset{\sim}{\nabla}e_{ip'})^T\right\}. \quad (3.17)$$

Equation (3.16) can hence be written as

$$\underset{\sim}{\nabla}(\underset{\sim}{\nabla}U(\underset{\sim}{\phi}(p'),p))^T D\underset{\sim}{\phi}(p') - (\text{sgn } M_{p'})q^2\left[\sum_{i\varepsilon K}\left(\frac{e_{ip'}}{M_{p'}}\right)^q\right]^{\frac{1}{q}-1}$$

$$\cdot\sum_{i\varepsilon K}\left\{\left(\frac{e_{ip'}}{M_{p'}}\right)^{q-1}\ln\left(\frac{e_{ip'}}{M_{p'}}\right)\underset{\sim}{\nabla}e_{ip'}\right\} = \underset{\sim}{0}\ , \quad (3.18)$$

where

$$D\underset{\sim}{\phi}(p') = \left[\frac{d\phi_1(p')}{dp'}\ \ \frac{d\phi_2(p')}{dp'}\ \ \cdots\ \ \frac{d\phi_n(p')}{dp'}\right]^T . \quad (3.19)$$

Under the stated assumptions, the matrix that multiplies $D\underset{\sim}{\phi}(p')$ in (3.18) has an inverse, so that

$$D\underset{\sim}{\phi}(p') = (\text{sgn } M_{p'}) \left\{ \underset{\sim}{\nabla}(\underset{\sim}{\nabla} U(\underset{\sim}{\phi}(p'),p))^T \right\}^{-1} q^2 \left(\sum_{i \in K} \left(\frac{e_{ip'}}{M_{p'}} \right)^q \right)^{\frac{1}{q} - 1}$$

$$\cdot \sum_{i \in K} \left\{ \left(\frac{e_{ip'}}{M_{p'}} \right)^{q-1} \ln \left(\frac{e_{ip'}}{M_{p'}} \right) \underset{\sim}{\nabla} e_{ip'} \right\} . \qquad (3.20)$$

Therefore, the derivatives of $\underset{\sim}{\phi}(p')$ exist for $1 > \frac{1}{p} > 0$ (or $1 < p < \infty$). If we differentiate (3.18) with respect to p', we shall find that the existence of the same inverse is required for $D^2\underset{\sim}{\phi}(p')$ to exist as required for $D\underset{\sim}{\phi}(p')$. In addition, $D^2\underset{\sim}{\phi}(p')$ requires the existence of the third partial derivatives of $e_i(\underset{\sim}{\phi})$ with respect to $\underset{\sim}{\phi}$. By continuing in this manner it should be possible to obtain explicitly all derivatives $D^k\underset{\sim}{\phi}(p')$ in terms of the derivatives $D^j\underset{\sim}{\phi}(p')$, $j=1,\ldots,k-1$, and partial derivatives of the functions $e_i(\underset{\sim}{\phi})$, $i=1,\ldots,m+1$, of degree up to k+1.

In order that the minimizing trajectory $\underset{\sim}{\phi}(p')$ be expanded in a Taylor series about $p' = 0$, we have to show that limiting derivatives exist at $p' = 0$. Let us first investigate the existence of $D\underset{\sim}{\phi}(p')$ as $p' \to 0$ or $p \to \infty$. Recall that the Hessian matrix is defined by (3.17); for very large values of p, we can approximate the matrix as

$$\underset{\sim}{\nabla}(\underset{\sim}{\nabla} U(\underset{\sim}{\phi}(p'),p))^T \cong q \left(\sum_{i \in K} \left(\frac{e_{ip'}}{M_{p'}} \right)^q \right)^{\frac{1}{q}-1} \sum_{i \in K} \left\{ \left(\frac{e_{ip'}}{M_{p'}} \right)^{q-2} \left(\frac{\underset{\sim}{\nabla} e_{ip'}}{M_{p'}} \right) (\underset{\sim}{\nabla} e_{ip'})^T \right\}$$

$$= p\, H_p \quad , \qquad (3.21)$$

where

$$H_p \triangleq (\operatorname{sgn} M_{p'})M_{p'}\ s_q(p') \sum_{i\varepsilon K} \left\{ \frac{\mu_i(p')}{e_{ip'}^2} \underset{\sim}{\nabla e}_{ip'} (\underset{\sim}{\nabla e}_{ip'})^T \right\} , \tag{3.22}$$

and

$$s_q(p') \triangleq \left(\sum_{i\varepsilon K} \left(\frac{e_{ip'}}{M_{p'}} \right)^q \right)^{\frac{1}{q}} , \tag{3.23}$$

$$\mu_i(p') \triangleq \frac{\left(\frac{e_{ip'}}{M_{p'}} \right)^q}{\sum_{i\varepsilon K} \left(\frac{e_{ip'}}{M_{p'}} \right)^q} . \tag{3.24}$$

H_p is an n×n matrix and for any nonzero n-component vector $\underset{\sim}{x}$,

$$\underset{\sim}{x}^T H_p \underset{\sim}{x}$$

$$= (\operatorname{sgn} M_{p'})M_{p'}\ s_q(p') \sum_{i\varepsilon K} \left\{ \frac{\mu_i(p')}{e_{ip'}^2} \underset{\sim}{x}^T \underset{\sim}{\nabla e}_{ip'} (\underset{\sim}{\nabla e}_{ip'})^T \underset{\sim}{x} \right\} . \tag{3.25}$$

Of interest is the positiveness of the terms $\underset{\sim}{x}^T \underset{\sim}{\nabla e}_{ip'} (\underset{\sim}{\nabla e}_{ip'})^T \underset{\sim}{x}$ in the summation. It follows that a necessary condition for $\underset{\sim}{x}^T H_p \underset{\sim}{x}$ to be positive is that for the gradient vectors $\underset{\sim}{\nabla e}_{ip'}$, $i\varepsilon\check{K}$, at least n of them are linearly independent, where

$$\check{K} \triangleq \{ i \,|\, e_i(\underset{\sim}{\phi}(0)) = M(\underset{\sim}{\phi}(0)),\ i\varepsilon I \} . \tag{3.26}$$

This ensures that the vector $\underset{\sim}{x}$ cannot be orthogonal to the n gradient vectors $\underset{\sim}{\nabla e}_{ip'}$ simultaneously, and at least one of the terms

$\underset{\sim}{x}^T \underset{\sim}{\nabla} e_{ip'} (\underset{\sim}{\nabla} e_{ip'})^T \underset{\sim}{x}$ will be positive. If the associated multipliers $\mu_i(p')$, $i\varepsilon\check{K}$, are positive, it is then sufficient for $\underset{\sim}{x}^T H_p \underset{\sim}{x}$ to be positive and H_p be positive definite and hence invertible.

Therefore, (3.20) becomes

$$\underset{\sim}{D\phi}(p') = (\text{sgn } M_{p'})\{p\, H_p\}^{-1} q^2 \left(\sum_{i\varepsilon K}\left(\frac{e_{ip'}}{M_{p'}}\right)^q\right)^{\frac{1}{q}-1} \cdot \sum_{i\varepsilon K}\left\{\left(\frac{e_{ip'}}{M_{p'}}\right)^{q-1} \ln\left(\frac{e_{ip'}}{M_{p'}}\right) \underset{\sim}{\nabla} e_{ip'}\right\}$$

$$= H_p^{-1}\left(\sum_{i\varepsilon K}\left(\frac{e_{ip'}}{M_{p'}}\right)^q\right)^{\frac{1}{q}-1} \sum_{i\varepsilon K}\left\{\left(\frac{e_{ip'}}{M_{p'}}\right)^{q-1} \ln\left(\frac{e_{ip'}}{M_{p'}}\right)^q \underset{\sim}{\nabla} e_{ip'}\right\}$$

$$= H_p^{-1}\left(\sum_{i\varepsilon K}\left(\frac{e_{ip'}}{M_{p'}}\right)^q\right)^{\frac{1}{q}} \sum_{i\varepsilon K}\left\{\frac{\left(\frac{e_{ip'}}{M_{p'}}\right)^q}{\sum_{i\varepsilon K}\left(\frac{e_{ip'}}{M_{p'}}\right)^q}\left[\ln \frac{\left(\frac{e_{ip'}}{M_{p'}}\right)^q}{\sum_{i\varepsilon K}\left(\frac{e_{ip'}}{M_{p'}}\right)^q}\right]\frac{M_{p'}}{e_{ip'}} \underset{\sim}{\nabla} e_{ip'}\right\}$$

$$= H_p^{-1} s_q(p') \sum_{i\varepsilon K}\left\{\mu_i(p')[\ln\mu_i(p')]\frac{M_{p'}}{e_{ip'}} \underset{\sim}{\nabla} e_{ip'}\right\} . \qquad (3.27)$$

Imposing optimality conditions [26], we observe that

$$\lim_{p'\to 0} s_q(p') = 1 , \qquad (3.28)$$

$$\lim_{p'\to 0} \mu_i(p') = v_i \begin{cases} = 0, & i\not\varepsilon\check{K} \\ \geq 0, & i\varepsilon\check{K} , \end{cases} \qquad (3.29)$$

$$\sum_{i\in\check{K}} v_i = 1 \ , \tag{3.30}$$

$$\lim_{p'\to 0} \frac{e_{ip'}}{M_{p'}} = 1 \ , \ i\varepsilon\check{K} \ , \tag{3.31}$$

and the gradient vectors $\nabla e_i(\phi(0))$, $i\varepsilon\check{K}$, are linearly dependent. Let us define $H_\infty = \lim_{p\to\infty} H_p$. Then, a necessary condition for H_∞ to be positive definite is that the set $\check{K}$ contains at least n+1 equal maxima and n of the associated gradient vectors $\nabla e_i(\phi(0))$ are linearly independent. A sufficient condition is that the multipliers v_i, $i\varepsilon\check{K}$, are positive.

The limiting value of $D\phi(p')$ at $p' = 0$ (or $p = \infty$) is therefore given by

$$D\phi(0) = \lim_{p'\to 0} D\phi(p')$$

$$= H_\infty^{-1} \sum_{i\varepsilon\check{K}} (v_i \ln v_i) \nabla e_i(\phi(0)) \ . \tag{3.32}$$

The existence of the higher order derivatives of $\phi(p')$ at $p' = 0$ may be derived in a similar manner.

To illustrate some of the ideas presented in this section, the Euclidean norms of $\phi(p')$ and $D\phi(p')$ of Example 1 for a sequence of p values, $p = 2$ to 2^{14}, were computed and tabulated in Table 3.5. Fig. 3.3 depicts the behaviour of $||\phi(\frac{1}{p})||_2$ and $||D\phi(\frac{1}{p})||_2$ as a function of $\frac{1}{p}$. We see that $||\phi(\frac{1}{p})||_2$ converges asymptotically to the value of $\sqrt{2}$

p	$\|\|\underset{\sim}{\phi}(\frac{1}{p})\|\|_2$	$\|\|D\underset{\sim}{\phi}(\frac{1}{p})\|\|_2$
2	1.30676	0.32544
2^2	1.36278	0.33159
2^3	1.38818	0.38769
2^4	1.40087	0.45442
2^5	1.40740	0.51015
2^6	1.41076	0.54797
2^7	1.41247	0.57044
2^8	1.41334	0.58276
2^9	1.41378	0.58923
2^{10}	1.41399	0.59255
2^{11}	1.41410	0.59423
2^{12}	1.41416	0.59507
2^{13}	1.41419	0.59550
2^{14}	1.41420	0.59571

Table 3.5 Euclidean norms of $\underset{\sim}{\phi}(\frac{1}{p})$ and $D\underset{\sim}{\phi}(\frac{1}{p})$.

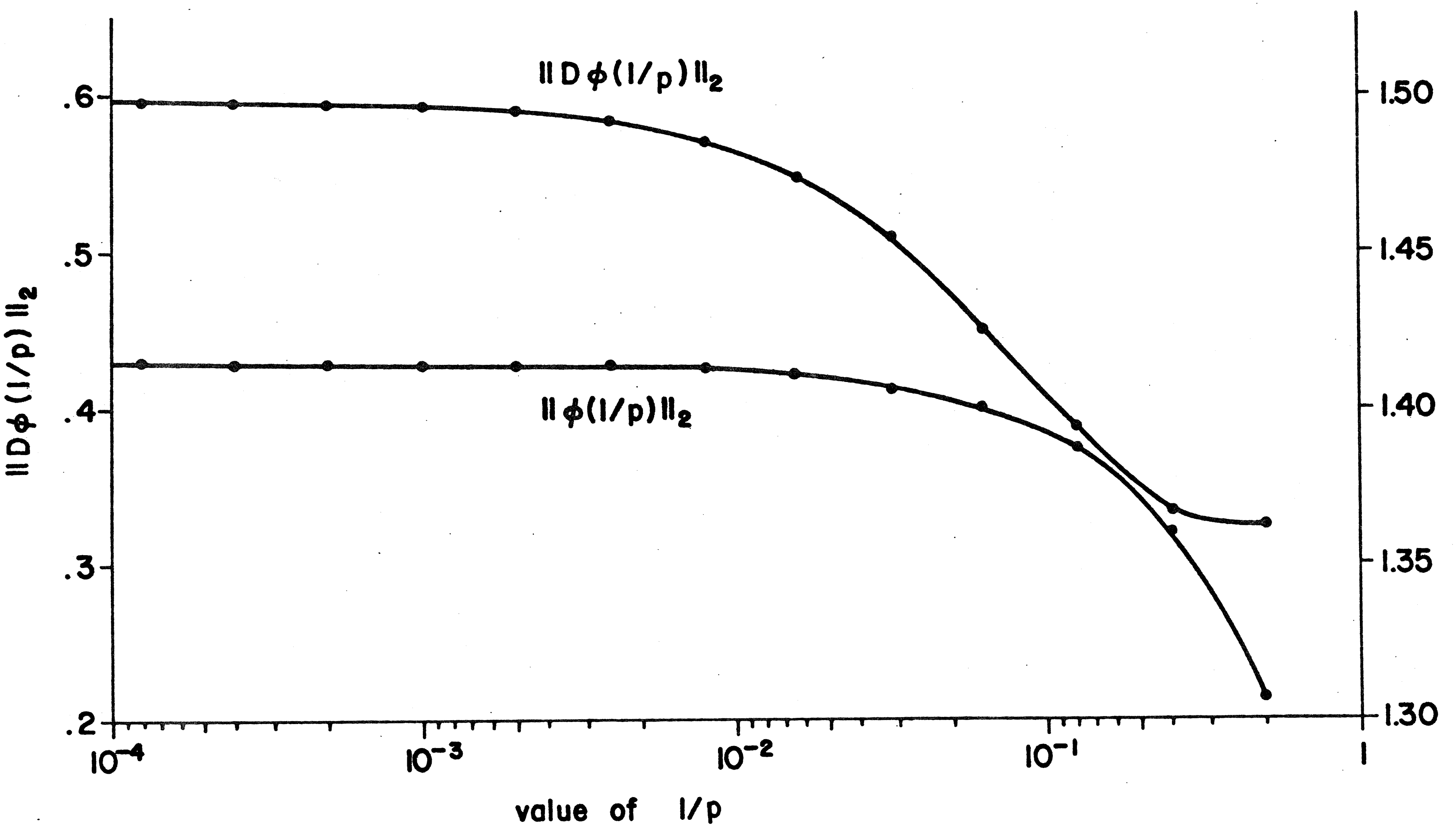

Figure 3.3 $||\underset{\sim}{\phi}(\frac{1}{p})||_2$ and $||D\underset{\sim}{\phi}(\frac{1}{p})||_2$ as a function of $\frac{1}{p}$.

(for the minimax minimum) and $||D\underset{\sim}{\phi}(\frac{1}{p})||_2$ is well defined.

With the existence of $D^i\underset{\sim}{\phi}(p')$, i = 1,...,k , at p' = 0, the results of section 2.4 can be applied directly if the parameter r is replaced by $\frac{1}{p}$. Estimates of the minimax optimum can be obtained by using the extrapolation formula (2.23).

3.5 Discussion

In the four examples considered, the performance of the extrapolation procedure in yielding the minimax solution is satisfactory. The order of estimates has been limited to three, though higher orders are possible. Computer storage requirements and accuracy considerations such as round-off error (which becomes critical for higher-order estimates) prompted our choice. Numerical experience indicates that the factor c by which p_i is increased (or r_i is reduced) is not crucial to convergence. In general, the faster the rate of increase (or decrease), the fewer are the number of minima required to obtain significant estimates of the solution values. Each minimum requires more computation to be reached than an increase (or decrease) at a slower rate. More minima are required to compute significant estimates in the later case. A practical range for c is 2 to 10. After all, the intention of applying extrapolation techniques is to avoid the necessity for calculating unconstrained minima for very large values of p (or small values of r).

Charalambous has recently devised a scheme [21] to predict the Kuhn-Tucker multipliers and the threshold value of the parameter α required in the Bandler-Charalambous minimax formulation. The scheme significantly accelerates the convergence of the Charalambous-Bandler minimax algorithms. In seeking a different approach to predict the threshold value of α, we find that the extrapolation formula (2.23) may be used to give us accurate estimates of the Kuhn-Tucker multipliers and hence the threshold value of α (see Appendix B).

As a matter of interest, Table 3.6 compares the results obtained from using the present least pth approach with extrapolation and those obtained by Charalambous [21]-[22] using acceleration techniques in his minimax algorithms in solving the Rosen-Suzuki problem. Though the three approaches are primarily least pth approximation methods, differences in objective formulations, approaches to acceleration and convergence criteria make it difficult to conclude which algorithm is the best.

Parameters	Least pth Approach with Extrapolation	Charalambous Algorithm [21]	Charalambous Algorithm [22]
ϕ_1	-0.0000002	0.000000	0.
ϕ_2	1.0000005	1.000000	0.999999
ϕ_3	1.9999999	2.000000	2.000001
ϕ_4	-1.0000002	-1.000000	-1.000000
$f(\underset{\sim}{\phi})$	-44.00000012	-44.00000000	-44.000003
Function evaluations	72	99	163

Table 3.6 Comparison of three least pth algorithms using acceleration techniques in solving the Rosen-Suzuki problem. The starting point is $\underset{\sim}{\phi}^0 = [0\ 0\ 0\ 0]^T$.

CHAPTER 4

CONCLUSIONS

Theoretical considerations and computational merits of applying an extrapolation technique in solving minimax problems and nonlinear programming problems using a sequence of least pth approximations or sequential unconstrained minimization techniques have been presented. Numerical results indicate that the new least pth approach using extrapolation is at least as efficient as or faster than most of the existing minimax algorithms which do not employ such an acceleration technique. A computer program package, called FLNLP2, incorporating the extrapolation technique and other recent optimization techniques is also developed. The program is capable of solving constrained or unconstrained optimization problems in general.

In Chapter 3, a preliminary analysis of the trajectory of least pth minima, which to the author's knowledge is a first attempt of its kind, leads to the confirmation of the extrapolation technique. It is felt that a rigorous analysis will reveal further useful information. In optimization problems, it is not uncommon to have symmetry or linear dependence in the problem variables. This may cause ill-conditioning and unnecessary computational effort for solving the problem. It is left for future investigation to apply a functional analysis of the first and second derivatives of the trajectory of least pth minima to determine and take into account the existence of symmetry or linear dependence in the early stages of the optimization process. The problem may then be redefined to have a better-behaved objective function and some computational effort may be saved. In reference [10], by enforcing symmetry in the design variables of the 7-element LC lowpass filter, a saving of about 40% in execution time is obtained.

The program FLNLP2 is written such that minimum effort is required of the user. A user is responsible for

(1) supplying in a main program the values and/or proper dimensioning of the parameters in the argument list and

(2) writing a service subroutine to define the objective function, the constraints and their respective partial derivatives.

Intermediate output and the final solution will be printed by FLNLP2 according to the user's discretion. As many optimization problems can be easily formulated as nonlinear programming problems, FLNLP2 should find a wide range of applications. The relatively small size of the program makes it ideal to be installed on a dedicated mini-computer that has moderate central memory storage, e.g., a PDP 11/45. However, the smaller word length of the mini-computer may require the extrapolation procedure to be performed in double precision to ensure accurate results. In solving the Rosen-Suzuki problem on the CDC 6400, there is no significant difference in accuracy between solutions obtained with single precision or double precision. Modifications to FLNLP2 can easily be made owing to the straightforward organization of the program. A logical modification will be an interactive version. This will offer greater flexibility to the user.

Throughout this thesis, extrapolation is used to accelerate convergence in solving minimax and nonlinear programming problems. Charalambous has recently presented new results in least pth optimization and nonlinear programming [21]-[22], using different approaches to acceleration. The overall efficiency of the present least pth approach with extrapolation may be enhanced by adopting some of the acceleration techniques proposed by Charalambous, or vice versa. Other acceleration techniques that appear to be worthwhile for consideration are:

(a) Applying the extrapolation formula (2.23) on the sequence of multipliers μ_i of (3.24) to determine which error functions are likely to be inactive at the solution of the minimax or nonlinear programming problem.

(b) Applying the extrapolation formula (2.23) and relation (3.21) to give the initial estimate of the Hessian matrix for the next cycle of optimization.

The present work has confirmed that extrapolation can be widely applied to accelerate convergence. Being a general program, FLNLP2 may not have solved the specific test examples in the most efficient possible way. However, other measures to improve efficiency have been suggested in this thesis.†

---

†The results improve over those presented previously [11] due to more efficient use of Fletcher's quasi-Newton subroutine. The unit matrix was used as the initial estimate of the Hessian matrix for each cycle of optimization. In the present work, except for the first cycle, the initial estimate is the Hessian matrix computed at the previous minimum, which is made possible by setting the parameter MODE to 3.

APPENDIX A

THE FLNLP2 PROGRAM*

A.1 Purpose

FLNLP2 is a package of subroutines for solving constrained optimization problems. That is, it minimizes a function

$$f \triangleq f(\underset{\sim}{x})$$

of n variables $\underset{\sim}{x}$ subject to the constraints

$$c_i(\underset{\sim}{x}) \geq 0 \ , \ i = 1,2,\ldots,n_c .$$

The technique proposed by Bandler and Charalambous [4] is used to transform the constrained optimization problem into the minimization of an unconstrained objective function. Practical least pth approximation is used to solve the resulting minimax problem. The user may use a very large value of p or a sequence of p values with extrapolation.

The package FLNLP2 is an updated version of the package FLNLP1 [23]. FLNLP2 differs from FLNLP1 by having the 1972 version of Fletcher's method and the option of an extrapolation technique. The program is currently limited to 100 constraints. To increase this limit, the COMMON statement WY3 has to be modified.

---

*The notation used is designed to appear consistent with the FØRTRAN names suggested to the user.

A.2 Argument List

SUBROUTINE FLNLP2 (N, NC, MM, IGK, X, G, H, W, EPS, XE, IH, IK, XB, IFINIS)

The arguments are as follows

N	An integer to be set to the number of variables ($N \geq 2$).
NC	An integer to be set to the number of constraints.
MM	An integer to be set to 1 if input data is to be read. Otherwise, set to zero.
IGK	An integer to be set to 1 if a gradient check by perturbation is desired. Otherwise, set to any other value. Also, gradient check is not performed when input data is not read.
X	A real array of N elements in which the current estimate of the solution is stored. An initial approximation must be set in X on entry. When the extrapolation procedure is used, an estimate of the next minimum in the sequence will be stored on exit of each cycle of optimization.
G	A real array of N elements in which the gradient vector corresponding to X above will be returned. When the extrapolation procedure is used, the optimal solution of each cycle of optimization will be returned in G on exit.
H	A real array of N*(N+1)/2 elements in which an estimate of the Hessian matrix is stored.
W	A real array of 4*N elements used as working space.
EPS	A real array of N elements to be set to the test quantities used in Fletcher's program.

XE — A real array of N*IK*(JORDER+1) elements in which different orders of estimates of the minimax solution are stored when extrapolation is used.

IH — An integer to be set to 1 if a single value of p is used. When a sequence of p values is used, IH should be set as the index of a DO loop that calls SUBROUTINE FLNLP2 IK times.

IK — An integer to be set to the maximum number of cycles of optimization. It corresponds to the number of p values when extrapolation is used.

XB — A real array of N elements in which the best estimate of the minimax solution currently available is stored.

IFINIS — An integer whose value will be equal to N when the convergence criterion for the estimates of the minimax solution is met.

A.3 Input Data

Parameters to be supplied as input data are defined as follows.

MAX — An integer to be set to the maximum number of iterations allowed.

IPT — An integer controlling printing of intermediate output. Printing occurs every |IPT| iterations and also on exit except when IPT is set to zero in which case intermediate output is suppressed.

ID — An integer to be set to 1 if input data is to be printed. Otherwise, set to zero.

EST	A real number to be set to the estimated minimum value of the artificial unconstrained objective function.
EPSC	A positive real number to be set to the error tolerance in the constraints.
AO	A positive real number to be set to the initial value of the parameter α used in formulating the unconstrained objective function.[§]
PO	A real number to be set to the value of p used in the least pth formulation or the initial value of p when a sequence of p values is used.
X(I) I=1,N	Starting values for the variables $x_1, x_2, \ldots, x_n$ defined in A.2.
EPS(I) I=1,N	As defined in A.2.
IEX	An integer to be set to 1 if the extrapolation procedure is used; otherwise set to zero.
JORDER	An integer to be set to the highest order of estimates used in extrapolation (JORDER $\leq$ IK-1); otherwise set to zero.
JPRINT	An integer to be set to 1 if the estimates of the minimax solution are to be printed; otherwise set to zero.
FACTOR	A positive real number to be set to the multiplying factor for p when a sequence of p values is used together with extrapolation; otherwise set to zero.

[§] When necessary, the parameter α will be increased by factors of 10 (maximum 5 times). The program will then stop if no feasible solution is found.

The input data is to be read in the following format:

CARD NO.	FORMAT	PARAMETERS
1	5I5	MAX, IPT, ID
2	5E16.8	EST, EPSC, AO, PO
As many as required	5E16.8	X(I), I = 1, N
As many as required	5E16.8	EPS(I), I = 1, N
Next	5I5	IEX, JORDER, JPRINT
Last	5E16.8	FACTOR

A.4 User Subroutines

The user has to supply the main program and a subroutine called FUNCT which defines the actual objective function, the constraint functions, and all first partial derivatives.

In the main program, the user has to supply the values and proper dimensioning for the parameters in the argument list of subroutine FLNLP2. The subroutine FLNLP2 needs to be called once when a single value of p is used. In using the extrapolation procedure, the subroutine FLNLP2 has to be called a number of times. This may be done, for example, by

```
      MM = 1
      IGK = 1
      DO 1  IH = 1 , IK
      CALL FLNLP2 (N, NC, MM, IGK, X, G, H, W, EPS, XE, IH, IK,
    1 XB, IFINIS)
```

```
      MM = 0
1     CONTINUE
```

The subroutine FUNCT should be written as follows:

```
      SUBROUTINE FUNCT (X, F, G, U)
      DIMENSION X(N), G(N), C(NC), GF(N), GC(N,NC), A(NT), TT(NT),
     1 TP(NT)
```

where

N = the number of independent variables (n)

NC = the number of inequality constraints (n_c)

NT = NC+1

F = $f(x_1,x_2,...,x_n)$ is the actual objective function

C(1) = $c_1(x_1,x_2,...,x_n)$

C(2) = $c_2(x_1,x_2,...,x_n)$

.
.
.

C(NC) = $c_{n_c}(x_1,x_2,...,x_n)$ are the inequality constraints

GF(1) = partial derivative of f w.r.t. x_1

.
.
.

GF(N) = partial derivative of f w.r.t. x_n

GC(1,1) = partial derivative of c_1 w.r.t. x_1

.
.
.

GC(N,NC) = partial derivative of c_{n_c} w.r.t. x_n

```
      CALL FMIMAX (N, NC, NT, F, G, GF, C, GC, U, A, TT, TP)
      RETURN
      END
```

If any other statements are necessary to define the actual objective function and the constraints, they may be added to this subroutine, e.g., function F may be defined in another subprogram which may be called by subroutine FUNCT.

A typical main program and subroutine FUNCT, the input deck, a printout of the input data and some final results for solving the Beale problem using a sequence of least pth approximations and extrapolation are given in Figs. A.1-A.3.

A.5 Other Subroutines

The following is a brief description of the subroutines called by FLNLP2.

FMIMAX	formulates the artificial unconstrained objective function and the necessary gradients.
GRDCHK	checks the gradient formulation by perturbation.
QUASIN	minimizes a function using the Fletcher unconstrained minimization program by quasi-Newton methods.
FINAL	outputs the optimal solution.
EXTRAP	performs extrapolation.

The overall structure of the program is shown in Fig. A.4.

A.6 Comments

The package was written to be used on the CDC 6400 computer. By itself, the package requires about 4,660 octal words of computer memory. The total amount of memory storage required when using the package to solve problems depends on the complexity of the subroutine FUNCT, the main program and the operating system of the computer. In solving the Beale problem, it took about 36,400 octal words of memory storage. The FTN compiler was used. The execution time for

```
      PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     MAIN PROGRAM
C
      DIMENSION X(3),G(3),H(6),W(12),EPS(3),XE(3,5,4),XB(3)
      N=3
      NC=4
      MM=1
      IGK=1
      IK=5
      DO 1 IH=1,IK
      CALL FLNLP2 (N,NC,MM,IGK,X,G,H,W,EPS,XE,IH,IK,XB,IFINIS)
      MM=0
      IF (IFINIS .EQ. N) CALL EXIT
1     CONTINUE
      STOP
      END

      SUBROUTINE FUNCT(X,F,G,U)

C     [illegible]

      [illegible]
      GC(2,2)=1.
      GC(3,2)=0.
      GC(1,3)=0.
      GC(2,3)=0.
      GC(3,3)=1.
      GC(1,4)=-1.
      GC(2,4)=-1.
      GC(3,4)=-2.
      CALL FMIMAX(3,4,5,F,G,GF,C,GC,U,A,TT,TP)
      RETURN
      END
C
C     INPUT DATA
C
 100     5     1
 0.0                 1.00000000E-06  1.0                 4.0
 1.0                 2.0             1.0
 1.00000000E-08      1.00000000E-08  1.00000000E-08
   1     3     1
 4.0
```

Figure A.1 Main program and subroutine FUNCT for the Beale problem. Input data is also shown.

INPUT DATA

```
NUMBER OF INDEPENDENT VARIABLES........................N =    3

MAXIMUM NUMBER OF ALLOWABLE ITERATIONS...............MAX = 100

INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS........IPT =    5

STARTING VALUE FOR VECTOR X(I).....................X( 1) =    .100000E+01
                                                   X( 2) =    .200000E+01
                                                   X( 3) =    .100000E+01

TEST QUANTITIES TO BE USED.......................EPS( 1) =    .100000E-07
                                                 EPS( 2) =    .100000E-07
                                                 EPS( 3) =    .100000E-07

ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZED..EST =   0.

THE MARGIN BY WHICH CONSTRAINTS MAY BE VIOLATED.....EPSC =    .100000E-05

INITIAL VALUE OF THE PARAMETER ALPHA..................A0 =    .100000E+01

HIGHEST ORDER OF ESTIMATES USED IN EXTRAPOLATION..JORDER =    3

MULTIPLYING FACTOR IN P VALUE.....................FACTOR =    .400000E+01
```

Figure A.2 Input data for the Beale problem.

```
IEXIT = 1
CRITERION FOR OPTIMUM (CHANGE IN VECTOR X .LT. EPS) HAS BEEN SATISFIED

OPTIMAL SOLUTION FOUND BY FLETCHER METHOD
-----------------------------------------

   ARTIFICIAL UNCONSTRAINED FUNCTION  U =   .11134106E+00
           ACTUAL OBJECTIVE FUNCTION  F =   .11123271E+00

   X( 1) =   .13335149E+01        G( 1) =  -.54275914E-08
   X( 2) =   .77765671E+00        G( 2) =  -.53990827E-08
   X( 3) =   .44414177E+00        G( 3) =  -.10739366E-07

   INEQUALITY CONSTRAINTS
   C( 1) =   .13335149E+01
   C( 2) =   .77765671E+00
   C( 3) =   .44414177E+00
   C( 4) =   .54481706E-03

          NUMBER OF FUNCTION EVALUATIONS =    4
     FINAL VALUE OF THE PARAMETER ALPHA =   .10000000E+01
              VALUE OF THE PARAMETER P =   .25600000E+03

ESTIMATES OF THE MINIMAX SOLUTION BY EXTRAPOLATION
--------------------------------------------------

ORDER  1
X( 1) =   .13333319E+01
X( 2) =   .77777875E+00
X( 3) =   .44444687E+00
ORDER  2
X( 1) =   .13333334E+01
X( 2) =   .77777775E+00
X( 3) =   .44444438E+00
ORDER  3
X( 1) =   .13333333E+01
X( 2) =   .77777778E+00
X( 3) =   .44444444E+00
          EXECUTION TIME IN SECONDS =   .039
```

Figure A.3 Results for the Beale problem.

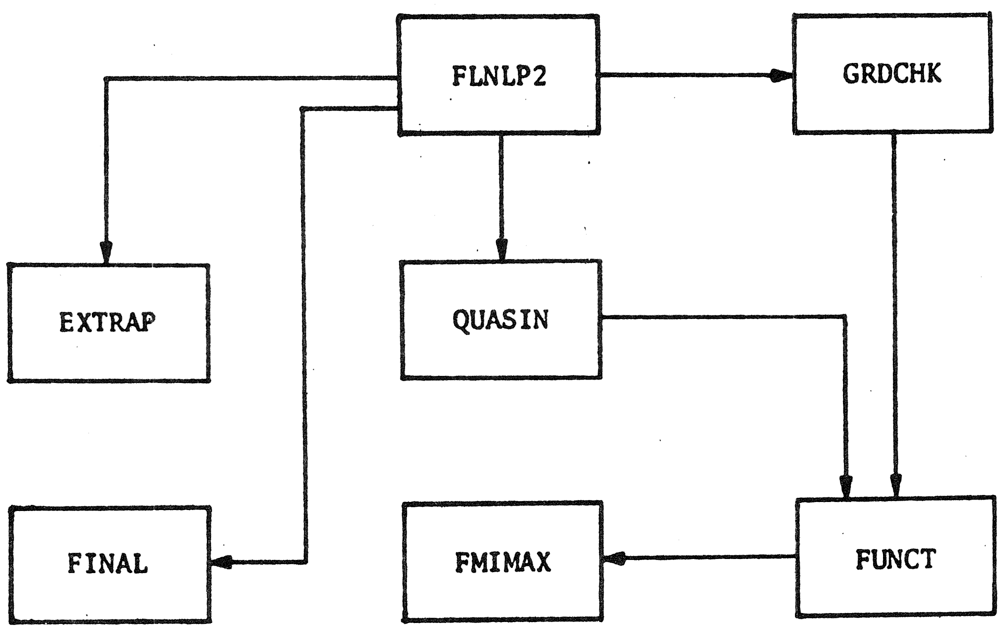

Figure A.4 Overall structure of FLNLP2.

the problem was about 0.9 second (central processor time) including printing of input and output.

The package is so organised that pertinent information of the optimization process can be obtained from the argument list of the subroutine FLNLP2. This allows the user to do some useful things in the main program, especially when using extrapolation, for example,

(i) In using the extrapolation procedure, we usually do not know how many cycles of optimization are required and the parameter IK may be set too large. The value of the parameter IFINIS may be used as a stopping criterion. When the accuracy in each element of XB (estimates of the minimax solution) is less than one hundred times of the accuracy in each element of X, the value of IFINIS becomes n. A statement as

```
IF (IFINIS.EQ.N) CALL EXIT
```

put inside the DO loop will serve the purpose.

(ii) System failure or time-limit may sometimes occur before the execution of the program is complete. Most of the information will be lost if it has not been saved. As a precaution, the user may at the end of each optimization cycle save the value of the array XE and the starting value of the next optimization cycle (which is stored in the array X) as punched output. Should restarting be necessary, the user simply reads in the value of the array XE obtained before the interruption, the starting value of $\underset{\sim}{x}$, some required input data and set the value of IH to the appropriate cycle number. The process will proceed as if nothing had happened.

Suppose time-limit occurred during execution of the fourth optimization cycle and we had saved relevant information of the previous three cycles. To restart the optimization process at the fourth cycle, the main program may contain the following statements:

```
      .
      .
      READ (5,2)    (XE(I,1,1), I = 1, N)
      READ (5,2)    (XE(I,2,1), I = 1, N)
      READ (5,2)    (XE(I,2,2), I = 1, N)
      READ (5,2)    (XE(I,3,1), I = 1, N)
      READ (5,2)    (XE(I,3,2), I = 1, N)
      READ (5,2)    (XE(I,3,3), I = 1, N)
2     FORMAT (5E16.8)
      MM = 1
      IGK = 0
      DO 1  IH = 4, IK
      CALL FLNLP2 (N, NC, MM, IGK, X, G, H, W, EPS, XE, IH, IK,
     1 XB, IFINIS)
      IF (IFINIS .EQ. N) CALL EXIT
      MM = 0
1     CONTINUE
      .
      .
```

The function of each statement is self-explanatory.

A.7 FORTRAN Listing for FLNLP2 Program

```
      SUBROUTINE FLNLP2 (N,NC,MM,IGK,X,G,H,W,EPS,XE,IH,IK,XB,IFINIS)
C
C     THIS SUBROUTINE SOLVES CONSTRAINED PROBLEMS USING LEAST PTH
C     APPROXIMATION AND EXTRAPOLATION
C
      COMMON /WY1/ IFN,KO
      COMMON /WY2/ ALFA,IA,IC,IM
      COMMON /WY4/ P,EPSC
      DIMENSION X(1), G(1), H(1), W(1), EPS(1), XE(N,IK,1), XB(1)
      IFINIS=0
      IF (MM.EQ.0) GO TO 2
      IEX=0
C
C     DATA INPUT
C
      READ (5,16) MAX,IPT,ID
      READ (5,24) EST,EPSC,AO,PO
      READ (5,24) (X(I),I=1,N)
      READ (5,24) (EPS(I),I=1,N)
      READ (5,16) IEX,JORDER,JPRINT
      READ (5,24) FACTOR
      P=PO
      ALFA=AO
      MODE=1
      IF (ID.EQ.0) GO TO 1
      WRITE (6,25)
      WRITE (6,26) N
      WRITE (6,27) MAX
      WRITE (6,28) IPT
      WRITE (6,29) X(1)
      WRITE (6,30) (I,X(I),I=2,N)
      WRITE (6,31) EPS(1)
      WRITE (6,32) (I,EPS(I),I=2,N)
      WRITE (6,33) EST
      WRITE (6,38) EPSC
      WRITE (6,17) AO
      IF (IEX.EQ.0) GO TO 1
      WRITE (6,18) JORDER
      WRITE (6,19) FACTOR
1     CONTINUE
C
C     GRADIENT CHECK
C
      IF (IGK.EQ.1) CALL GRDCHK (N,X,G,W)
2     CONTINUE
      P=PO
      IF (IPT.EQ.0) GO TO 3
      WRITE (6,34)
      WRITE (6,35)
      WRITE (6,36)
3     IM=0
      IT=0
      CALL SECOND (T1)
4     IC=0
C
C     MINIMIZATION
C
      CALL QUASIN (N,X,U,G,H,W,EST,EPS,MODE,MAX,IPT,IEXIT)
      IC=1
      IM=1
      MODE=3
      CALL FUNCT (X,F,G,U)
      IF (IA.EQ.0.OR.IEXIT.EQ.3) GO TO 6
      IF (IT.EQ.5) GO TO 5
      ALFA=ALFA*10.0
      IT=IT+1
      GO TO 4
5     KO=0
      PRINT 15
6     CALL SECOND (T2)
C
C     SOLUTION OUTPUT
C
      CALL FINAL (N,X,F,G,NC,U)
      PRINT 20, P
```

```
      IF (IT.EQ.5) CALL EXIT
      IF (IEX.EQ.0) GO TO 14
      DO 7 I=1,N
      G(I)=X(I)
    7 CONTINUE
C
C     EXTRAPOLATION
C
      CALL EXTRAP (N,X,XE,IH,IK,FACTOR,XB,JORDER)
      J1=JORDER+1
      IHH=IH-1
      IF (JPRINT.EQ.0.OR.IH.EQ.1) GO TO 10
      IJ=J1
      IF (IH.LE.J1) IJ=IH
      PRINT 21
      DO 9 L=2,IJ
      L1=L-1
      PRINT 22, L1
      DO 8 J=1,N
      PRINT 23, J,XE(J,IH,L)
    8 CONTINUE
    9 CONTINUE
   10 IF (IH.LT.3) GO TO 14
      IF (IH.GT.J1) GO TO 12
      DO 11 I=1,N
      IF (ABS(XE(I,IH,IH)-XE(I,IHH,IHH)).LT.100.*EPS(I)) IFINIS=IFINIS+1
   11 CONTINUE
      GO TO 14
   12 DO 13 I=1,N
      IF (ABS(XE(I,IH,J1)-XE(I,IHH,J1)).LT.100.*EPS(I)) IFINIS=IFINIS+1
   13 CONTINUE
   14 T=T2-T1
      WRITE (6,37) T
      PO=P*FACTOR
      RETURN
C
C
C
   15 FORMAT (*0THE PARAMETER ALPHA HAS BEEN INCREASED 5 TIMES, NO FEASI
     1BLE SOLUTION HAS BEEN FOUND.*)
   16 FORMAT (5I5)
   17 FORMAT (1H0/1H ,*INITIAL VALUE OF THE PARAMETER ALPHA*,18(*.*),*AO
     1 =*,F14.6)
   18 FORMAT (1H0/1H ,*HIGHEST ORDER OF ESTIMATES USED IN EXTRAPOLATION*
     1,*..JORDER =*,I4)
   19 FORMAT (1H0/1H ,*MULTIPLYING FACTOR IN P VALUE*,21(*.*),*FACTOR =*
     1,E14.6)
   20 FORMAT (1H0,15X,*VALUE OF THE PARAMETER P =*,E16.8)
   21 FORMAT (1H0/1H0,*ESTIMATES OF THE MINIMAX SOLUTION BY EXTRAPOLATIO
     1N*/1H ,50(*-*)/)
   22 FORMAT (1H0,*ORDER*,I3)
   23 FORMAT (1H0,*X(*,I2,*) =*,E16.8)
   24 FORMAT (5E16.8)
   25 FORMAT (1H1,*INPUT DATA*/,1H ,10(*-*),//)
   26 FORMAT (1H0,*NUMBER OF INDEPENDENT VARIABLES*,24(*.*),*N =*,I4,/)
   27 FORMAT (1H0,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS*,15(*.*),*MAX
     1=*,I4,/)
   28 FORMAT (1H0,*INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS*,8(*.*)
     1,*IPT =*,I4,/)
   29 FORMAT (1H0,*STARTING VALUE FOR VECTOR X(I)*,21(*.*),*X( 1) =*,E14
     1.6)
   30 FORMAT (1H0,51X,*X(*,I2,*) =*,E14.6)
   31 FORMAT (1H0/1H ,*TEST QUANTITIES TO BE USED*,23(*.*),*EPS( 1) =*,E
     114.6)
   32 FORMAT (1H0,49X,*EPS(*,I2,*) =*,E14.6)
   33 FORMAT (1H0/1H ,*ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZ
     1ED*,2(*.*),*EST =*,E14.6)
   34 FORMAT (1H1)
   35 FORMAT (1H0,*OPTIMIZATION BY FLETCHER METHOD*/,1H ,31(*-*),/)
   36 FORMAT (1H0,*ITER.*,2X,*FUNCT.*,6X,*ALPHA*,8X,*OBJECTIVE*,6X,*VARI
     1ABLE*,7X,*GRADIENT*/1H0,1X,*NO.*,3X,*EVALU.*,19X,*FUNCTION*,6X,*VE
     2CTOR X(I)*,4X,*VECTOR G(I)*,/)
   37 FORMAT (1H0,14X,*EXECUTION TIME IN SECONDS =*,F7.3)
   38 FORMAT (1H0/1H ,*THE MARGIN BY WHICH CONSTRAINTS MAY BE VIOLATED*,
     15(*.*),*EPSC =*,E14.6)
      END
```

```
      SUBROUTINE EMIMAX (N,NC,NT,F,G,GF,C,GC,U,A,TT,TP)
C
C     THIS SUBROUTINE TRANSFORMS THE CONSTRAINED PROBLEM INTO AN
C     UNCONSTRAINED OBJECTIVE USING THE BANDLER-CHARALAMBOUS TECHNIQUE
C
      COMMON /WY2/ ALFA,IA,IC,IM
      COMMON /WY3/ PC(100)
      COMMON /WY4/ P,EPSC
      DIMENSION GF(N), C(NC), GC(N,NC), G(N), A(NT), TT(NT), TP(NT)
      Q=P
      AE=0.0
      IA=0
      IF (NC.EQ.0.OR.ALFA.EQ.0.0) GO TO 12
      FA=F/ALFA
      DO 1 I=1,NC
      A(I)=FA-C(I)
    1 CONTINUE
      AM=A(1)
      A(NT)=FA
      DO 2 I=2,NT
      AM=AMAX1(AM,A(I))
    2 CONTINUE
      IF (AM.LE.0.0) Q=-Q
      SUM1=0.0
      DO 6 I=1,NT
      IF (AM) 5,3,4
    3 AE=1.E-10
      GO TO 5
    4 IF (A(I).LE.0.0) GO TO 6
    5 TT(I)=(A(I)-AE)/(AM-AE)
      TP(I)=TT(I)**Q
      SUM1=SUM1+TP(I)
    6 CONTINUE
      SUMT=ALFA*SUM1**(1./Q)
      U=(AM-AE)*SUMT
      DO 11 I=1,N
      XX=GF(I)/ALFA
      SUM2=0.0
      DO 10 J=1,NT
      IF (AM) 8,8,7
    7 IF (A(J).LE.0.0) GO TO 10
    8 YY=TP(J)/TT(J)
      ZZ=XX*YY
      IF (J.EQ.NT) GO TO 9
      SUM2=SUM2+ZZ-YY*GC(I,J)
      GO TO 10
    9 SUM2=SUM2+ZZ
   10 CONTINUE
      G(I)=(SUMT*SUM2)/SUM1
   11 CONTINUE
      GO TO 14
   12 U=F
      DO 13 I=1,N
      G(I)=GF(I)
   13 CONTINUE
   14 IF (IC.EQ.0.OR.NC.EQ.0) GO TO 16
      DO 15 I=1,NC
      PC(I)=C(I)
      CT=C(I)+EPSC
      IF (CT.LT.0.0) IA=1
   15 CONTINUE
   16 RETURN
      END
```

```
      SUBROUTINE GRDCHK (N,X,G,W)
C
C     THIS SUBROUTINE PERFORMS GRADIENT CHECK
C
      DIMENSION X(N), G(N), W(N)
      JJJ=0
      CALL FUNCT (X,F,G,U)
      WRITE (6,3)
      WRITE (6,4)
      DO 1 I=1,N
      Z=X(I)
      DX=1.E-4*X(I)
      IF (ABS(DX).LT.1.E-10) DX=1.E-10
      X(I)=Z+DX
      CALL FUNCT (X,F,W,U2)
      X(I)=Z-DX
      CALL FUNCT (X,F,W,U1)
      Y=0.5*(U2-U1)/DX
      X(I)=Z
      IF (ABS(Y).LT.1.E-14) Y=1.E-14
      IF (ABS(G(I)).LT.1.E-14) G(I)=1.E-14
      YP=ABS((Y-G(I))/Y)*100.0
      WRITE (6,5) G(I),Y,YP
      IF (YP.GT.10.0) JJJ=1
    1 CONTINUE
      IF (JJJ.EQ.1) GO TO 2
      WRITE (6,6)
      RETURN
    2 WRITE (6,7)
      CALL EXIT
C
C
C
    3 FORMAT (1H1,/1H ,*GRADIENT CHECK AT STARTING POINT*/1H ,32(*-*))
    4 FORMAT (//,1H0,5X,*ANALYTICAL GRADIENTS*,5X,*NUMERICAL GRADIENTS*
     1,5X,*PERCENTAGE ERROR*)
    5 FORMAT (1H0,8X,E14.6,10X,E14.6,8X,E14.6)
    6 FORMAT (1H0,///1H ,*GRADIENTS ARE O.K.*)
    7 FORMAT (1H0,///1H ,*YOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRADI
     1ENTS ARE INCORRECT*/1H0,*PLEASE CHECK IT AGAIN*)
      END
```

```
      SUBROUTINE QUASIN (N,X,U,G,H,W,EST,EPS,MODE,MAX,IPT,IEXIT)
C
C     THIS SUBROUTINE IS THE FLETCHER (1972) METHOD OF MINIMIZATION
C
      DIMENSION X(1), G(1), H(1), W(1), EPS(1)
      COMMON /WY1/ IFN,KO
      COMMON /WY2/ ALFA,IA,IC,IM
      KO=0
      IF (IM.EQ.1) GO TO 1
      ITN=0
      IFN=1
1     CONTINUE
      NP=N+1
      N1=N-1
      NN=N*NP/2
      IS=N
      IU=N
      IV=N+N
      IB=IV+N
      IEXIT=0
      IF (MODE.EQ.3) GO TO 7
      IF (MODE.EQ.2) GO TO 4
      IJ=NN+1
      DO 3 I=1,N
      DO 2 J=1,I
      IJ=IJ-1
      H(IJ)=0.
2     CONTINUE
      H(IJ)=1.
3     CONTINUE
      GO TO 7
4     CONTINUE
      IJ=1
      DO 6 I=2,N
      Z=H(IJ)
      IF (Z.LE.0.) RETURN
      IJ=IJ+1
      I1=IJ
      DO 6 J=I,N
      ZZ=H(IJ)
      H(IJ)=H(IJ)/Z
      JK=IJ
      IK=I1
      DO 5 K=I,J
      JK=JK+NP-K
      H(JK)=H(JK)-H(IK)*ZZ
      IK=IK+1
5     CONTINUE
6     IJ=IJ+1
      IF (H(IJ).LE.0.) RETURN
7     CONTINUE
      IJ=NP
      DMIN=H(1)
      DO 8 I=2,N
      IF (H(IJ).GE.DMIN) GO TO 8
      DMIN=H(IJ)
8     IJ=IJ+NP-I
      IF (DMIN.LE.0.) RETURN
      Z=EST
      CALL FUNCT (X,F,G,U)
      DF=U-EST
      IF (DF.LE.0.0) DF=1.0
9     CONTINUE
      IF (IPT.EQ.0) GO TO 10
      IF (MOD(ITN,IPT).NE.0) GO TO 10
      PRINT 45, ITN,IFN,ALFA,U,((X(I),G(I)),I=1,N)
10    CONTINUE
      ITN=ITN+1
      W(1)=-G(1)
      DO 12 I=2,N
      IJ=I
      I1=I-1
      Z=-G(I)
      DO 11 J=1,I1
      Z=Z-H(IJ)*W(J)
```

```
      IJ=IJ+N-J
11    CONTINUE
      W(I)=Z
12    CONTINUE
      W(IS+N)=W(N)/H(NN)
      IJ=NN
      DO 14 I=1,N1
      IJ=IJ-1
      Z=0.
      DO 13 J=1,I
      Z=Z+H(IJ)*W(IS+NP-J)
      IJ=IJ-1
13    CONTINUE
      W(IS+N-I)=W(N-I)/H(IJ)-Z
14    CONTINUE
      GS=0.
      DO 15 I=1,N
      GS=GS+W(IS+I)*G(I)
15    CONTINUE
      IEXIT=2
      IF (GS.GE.0.) GO TO 37
      GSO=GS
      ALPHA=-2.*DF/GS
      IF (ALPHA.GT.1.) ALPHA=1.
      DF=U
      TOT=0.
16    CONTINUE
      IEXIT=3
      IF (ITN.EQ.MAX) GO TO 37
      ICON=0
      IEXIT=1
      DO 17 I=1,N
      Z=ALPHA*W(IS+I)
      IF (ABS(Z).GE.EPS(I)) ICON=1
      X(I)=X(I)+Z
17    CONTINUE
      CALL FUNCT (X,F,W,UY)
      IFN=IFN+1
      GYS=0.
      DO 18 I=1,N
      GYS=GYS+W(I)*W(IS+I)
18    CONTINUE
      IF (UY.GE.U) GO TO 19
      IF (ABS(GYS/GSO).LE..9) GO TO 21
      IF (GYS.GT.0.) GO TO 19
      TOT=TOT+ALPHA
      Z=10.
      IF (GS.LT.GYS) Z=GYS/(GS-GYS)
      IF (Z.GT.10.) Z=10.
      ALPHA=ALPHA*Z
      U=UY
      GS=GYS
      GO TO 16
19    CONTINUE
      DO 20 I=1,N
      X(I)=X(I)-ALPHA*W(IS+I)
20    CONTINUE
      IF (ICON.EQ.0) GO TO 37
      Z=3.*(U-UY)/ALPHA+GYS+GS
      ZZ=SQRT(Z*Z-GS*GYS)
      GZ=GYS+ZZ
      Z=1.-(GZ-Z)/(ZZ+GZ-GS)
      ALPHA=ALPHA*Z
      GO TO 16
21    CONTINUE
      ALPHA=TOT+ALPHA
      U=UY
      IF (ICON.EQ.0) GO TO 35
      DF=DF-U
      DGS=GYS-GSO
      LINK=1
      IF (DGS+ALPHA*GSO.GT.0.) GO TO 23
      DO 22 I=1,N
      W(IU+I)=W(I)-G(I)
22    CONTINUE
```

```
      SIG=1./(ALPHA*DGS)
      GO TO 30
   23 CONTINUE
      ZZ=ALPHA/(DGS-ALPHA*GS0)
      Z=DGS*ZZ-1.
      DO 24 I=1,N
      W(IU+I)=Z*G(I)+W(I)
   24 CONTINUE
      SIG=1./(ZZ*DGS*DGS)
      GO TO 30
   25 CONTINUE
      LINK=2
      DO 26 I=1,N
      W(IU+I)=G(I)
   26 CONTINUE
      IF (DGS+ALPHA*GS0.GT.0.) GO TO 27
      SIG=1./GS0
      GO TO 30
   27 CONTINUE
      SIG=-ZZ
      GO TO 30
   28 CONTINUE
      DO 29 I=1,N
      G(I)=W(I)
   29 CONTINUE
      GO TO 9
   30 CONTINUE
      W(IV+1)=W(IU+1)
      DO 32 I=2,N
      IJ=I
      I1=I-1
      Z=W(IU+I)
      DO 31 J=1,I1
      Z=Z-H(IJ)*W(IV+J)
      IJ=IJ+N-J
   31 CONTINUE
      W(IV+I)=Z
   32 CONTINUE
      IJ=1
      DO 33 I=1,N
      IVI=IV+I
      IBI=IB+I
      Z=H(IJ)+SIG*W(IVI)*W(IVI)
      IF (Z.LE.0.) Z=DMIN
      IF (Z.LT.DMIN) DMIN=Z
      H(IJ)=Z
      W(IBI)=W(IVI)*SIG/Z
      SIG=SIG-W(IBI)*W(IBI)*Z
      IJ=IJ+NP-I
   33 CONTINUE
      IJ=1
      DO 34 I=1,N1
      IJ=IJ+1
      I1=I+1
      DO 34 J=I1,N
      W(IU+J)=W(IU+J)-H(IJ)*W(IV+I)
      H(IJ)=H(IJ)+W(IB+I)*W(IU+J)
   34 IJ=IJ+1
      GO TO (25,28), LINK
   35 CONTINUE
      DO 36 I=1,N
      G(I)=W(I)
   36 CONTINUE
   37 CONTINUE
      IF (IEXIT.EQ.1) KO=1
      IF (IPT.EQ.0) GO TO 38
      PRINT 45, ITN,IFN,ALFA,U,((X(I),G(I)),I=1,N)
   38 IF (IEXIT.EQ.0) GO TO 39
      GO TO 40
   39 PRINT 46, IEXIT
      GO TO 44
   40 GO TO (41,42,43), IEXIT
   41 PRINT 47, IEXIT
      GO TO 44
```

```
42    PRINT 48, IEXIT
      GO TO 44
43    PRINT 49, IEXIT
44    CONTINUE
      RETURN
C
C
C
45    FORMAT (1H ,I3,5X,I3,6X,E10.3,1X,E14.6,1X,80(E14.6,1X,E14.6/44X))
46    FORMAT (1H1,*IEXIT =*,I2/1H0,*THE ESTIMATE OF THE HESSIAN MATRIX I
     1S NOT POSITIVE DEFINITE*)
47    FORMAT (1H1,*IEXIT =*,I2/1H0,*CRITERION FOR OPTIMUM (CHANGE IN VEC
     1TOR X .LT. EPS) HAS BEEN SATISFIED*)
48    FORMAT (1H1,*IEXIT =*,I2/1H0,*EPS CHOSEN IS TOO SMALL*)
49    FORMAT (1H1,*IEXIT =*,I2/1H0,*MAXIMUM NUMBER OF ALLOWABLE ITERATIO
     1NS HAS BEEN REACHED*)
      END
```

```
      SUBROUTINE EXTRAP (N,X,XF,IH,IK,FACTOR,XB,JORDER)
C
C     THIS SUBROUTINE PERFORMS EXTRAPOLATION
C
      DIMENSION X(1), XF(N,IK,1), XB(1)
      I=IH
      II=I+1
      DO 1 J=1,N
      XF(J,I,1)=X(J)
1     CONTINUE
      IF (I.LT.2) GO TO 11
      IF (I.GT.JORDER) GO TO 2
      IJ=I
      GO TO 3
2     IJ=JORDER+1
C
C     ESTIMATES OF THE ULTIMATE SOLUTION
C
3     DO 5 L=2,IJ
      LL=L-1
      S=FACTOR**LL
      DO 4 J=1,N
      XF(J,I,L)=(S*XF(J,I,LL)-XF(J,I-1,LL))/(S-1.0)
4     CONTINUE
5     CONTINUE
      DO 6 J=1,N
      XB(J)=XF(J,I,IJ)
6     CONTINUE
      IF (I.EQ.IK) RETURN
C
C     ESTIMATE OF THE NEXT STARTING POINT
C
      DO 7 J=1,N
      XF(J,II,IJ)=XF(J,I,IJ)
7     CONTINUE
      DO 9 K=2,IJ
      L=IJ+1-K
      SS=FACTOR**L
      DO 8 J=1,N
```

```
      XF(J,II,L)=((SS-1.0)*XF(J,II,L+1)+XE(J,I,L))/SS
8     CONTINUE
9     CONTINUE
      DO 10 J=1,N
      X(J)=XF(J,II,1)
10    CONTINUE
      RETURN
11    DO 12 J=1,N
      XB(J)=XE(J,I,1)
12    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FINAL (N,X,F,G,NC,U)
C
C     THIS SUBROUTINE OUTPUTS THE OPTIMAL SOLUTION
C
      DIMENSION X(N), G(N)
      COMMON /WY1/ IEN,KO
      COMMON /WY2/ ALFA,IA,IC,IM
      COMMON /WY3/ PC(100)
      IF (KO.EQ.0) GO TO 1
      WRITE (6,4)
      GO TO 2
1     WRITE (6,5)
2     WRITE (6,6) U
      WRITE (6,9) F
      WRITE (6,7) (I,X(I),I,G(I),I=1,N)
      IF (NC.EQ.0) GO TO 3
      WRITE (6,10)
      WRITE (6,11) (I,PC(I),I=1,NC)
3     WRITE (6,8) IEN
      WRITE (6,12) ALFA
      RETURN
C
C
C
4     FORMAT (1H0,/1H0,*OPTIMAL SOLUTION FOUND BY FLETCHER METHOD*/1H ,4
     11(*-*))
5     FORMAT (1H0,/1H0,*RESULTS FOUND BY FLETCHER METHOD AT LAST ITERATI
     1ON*/1H ,50(*-*))
6     FORMAT (1H0,//4X,*ARTIFICIAL UNCONSTRAINED FUNCTION  U =*,E16.8)
7     FORMAT (1H0,3X,*X(*,I2,*) =*,E16.8,8X,*G(*,I2,*) =*,E16.8)
8     FORMAT (1H0,/1H ,9X,*NUMBER OF FUNCTION EVALUATIONS =*,I5)
9     FORMAT (1H0,11X,*ACTUAL OBJECTIVE FUNCTION  F =*,E16.8/)
10    FORMAT (1H0,/1H ,3X,*INEQUALITY CONSTRAINTS*)
11    FORMAT (1H0,3X,*C(*,I2,*) =*,E16.8)
12    FORMAT (1H0,5X,*FINAL VALUE OF THE PARAMETER ALPHA =*,E16.8)
      END
```

APPENDIX B

ESTIMATION OF KUHN-TUCKER MULTIPLIERS

The various quantities appearing in Table B.1 for the Rosen-Suzuki problem were generated by the following procedure:

1. Set the value of α to 1.
2. Minimize the objective function (3.1) for the sequence of p values of 4, 16, 64, 256, 1024.
3. At each least pth minimum, calculate the quantities μ_i of (3.24).
4. With the five sets of μ_i's, use the extrapolation formula (2.23) to estimate the v_i's of (3.29) and calculate the Kuhn-Tucker multipliers by using the relation $\check{u}_i = \alpha v_i$, $i = 1,2,3$.
5. Increase the value of α by one (until $\alpha = 10$); each time repeat steps 2 to 4.

Optimality requires that

$$\sum_{i=1}^{3} \frac{\check{u}_i}{\alpha} < 1 .$$

From the results shown in Table B.1, we see that the threshold value of α for the Rosen-Suzuki problem is 3.

α	v_1	v_2	v_3	v_4	$\sum_{i=1}^{4} v_i$	Kuhn-Tucker Multipliers $\check{u}_i$	$\check{u}_2$	$\check{u}_3$	$\sum_{i=1}^{3} \check{u}_i$
1	1.0064	0.0023	-0.0086	-0.0000	1.0001	1.0064	0.0023	-0.0086	1.0001
2	0.7317	0.0003	0.2680	-0.0000	1.0000	1.4635	0.0006	0.5359	2.0000
3	0.3355	0.0001	0.6612	0.0032	1.0000	1.0065	0.0002	1.9836	2.9903
4	0.2500	0.0000	0.4999	0.2500	1.0000	1.0001	0.0000	1.9998	2.9999
5	0.2000	-0.0000	0.4000	0.4000	1.0000	1.0000	-0.0000	1.9999	3.0000
6	0.1667	-0.0000	0.3333	0.5000	1.0000	1.0000	-0.0000	2.0000	3.0000
7	0.1429	-0.0000	0.2857	0.5714	1.0000	1.0000	-0.0000	2.0000	3.0000
8	0.1250	-0.0000	0.2500	0.6250	1.0000	1.0000	-0.0000	2.0000	3.0000
9	0.1111	-0.0000	0.2222	0.6667	1.0000	1.0000	-0.0000	2.0000	3.0000
10	0.1000	-0.0000	0.2000	0.7000	1.0000	1.0000	-0.0000	2.0000	3.0000

Table B.1 Kuhn-Tucker multipliers for the Rosen-Suzuki problem.

REFERENCES

[1] A. V. Fiacco and G. P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques. New York: Wiley, 1968.

[2] A. V. Fiacco and G. P. McCormick, "Extension of SUMT for nonlinear programming: Equality constraints and extrapolation", Management Sci., vol. 12, 1966, pp. 816-828.

[3] F. A. Lootsma, "Extrapolation in logarithmic programming", Philips Res. Repts. 23, 1968, pp. 108-116.

[4] J. W. Bandler and C. Charalambous, "Nonlinear programming using minimax techniques", J. Optimization Theory and Applications, vol. 13, June 1974, pp. 607-619.

[5] J. W. Bandler and C. Charalambous, "Practical least pth optimization of networks", IEEE Trans. Microwave Theory Tech., vol. MTT-20, Dec. 1972, pp. 834-840.

[6] R. Fletcher, "FORTRAN subroutines for minimization by quasi-Newton methods", Atomic Energy Research Establishment, Harwell, Berkshire, England, Report AERE-R7125, 1972.

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

[8] J. W. Bandler, "Optimization methods for computer-aided design", IEEE Trans. Microwave Theory Tech., vol. MTT-17, Aug. 1969, pp. 533-552.

[9] R. Seviora, M. Sablatash, and J. W. Bandler, "Least pth and minimax objectives for automated network design", Electron. Lett.,

vol. 6, 1970, pp. 14-15.

[10] J. W. Bandler, W. Y. Chu and J. R. Popović, "Efficient, interactive semi-automated optimization of models and designs", Proc. Symposium on Computers, Electronics and Control (Calgary, Alberta, Canada, May 1974), paper VI-4.

[11] J. W. Bandler and W. Y. Chu, "Computational merits of extrapolation in least pth approximation and nonlinear programming", Proc. 12th Annual Allerton Conference on Circuit and System Theory (Monticello, Ill., Oct. 1974).

[12] R. Frisch, The Logarithmic Potential Method for Solving Linear Programming Problems, Memorandum of the University Institute of Economics, Oslo, 1955.

[13] F. A. Lootsma, "Logarithmic programming: a method of solving nonlinear programming problems", Philips Res. Repts. 22, 1967, pp. 329-344.

[14] C. W. Carroll, "The created response surface technique for optimizing nonlinear restrained systems", Operat. Res. 9, 1961, pp. 169-184.

[15] J. Kowalik, "Nonlinear programming procedures and design optimization", Acta Polytech. Scand. 13, Trondheim, 1966.

[16] R. Fletcher and A. P. McCann, "Acceleration techniques for nonlinear programming", in Optimization, Ed. R. Fletcher, London: Academic Press, 1969, pp. 203-214.

[17] T. M. Apostol, Mathematical Analysis, Reading: Addison-Wesley, 1957, p. 147.

SOC-71

EXTRAPOLATION IN LEAST pTH APPROXIMATION AND NONLINEAR PROGRAMMING

W.Y. Chu

December 1974, No. of Pages: 69

Revised:

Key Words: Nonlinear programming, least pth approximation, extrapolation, minimax approximation, penalty function methods

Abstract: Theoretical considerations and computational merits of applying an extrapolation technique in solving minimax problems and nonlinear programming problems using a sequence of least pth approximations or sequential unconstrained minimization techniques is presented. Numerical results indicate that the new least pth approach using extrapolation is competitive with other established minimax algorithms. An efficient, user-oriented computer program, called FLNLP2, incorporating the extrapolation technique and other recent optimization techniques is also developed. The program is capable of solving constrained or unconstrained general optimization problems and is readily applicable to circuit design problems. The extrapolation technique has been illustrated in solving the Beale problem, the Rosen-Suzuki problem, an LC lowpass filter design problem and other test examples.

Description: M. Eng. Thesis.
Contains Fortran listing, user's manual.
Source deck available for $50.00.

Related Work: SOC-2, SOC-12, SOC-15, SOC-29, SOC-42, SOC-78, SOC-80, SOC-84, SOC-93, SOC-108, SOC-113, SOC-151.

Price: $30.00.