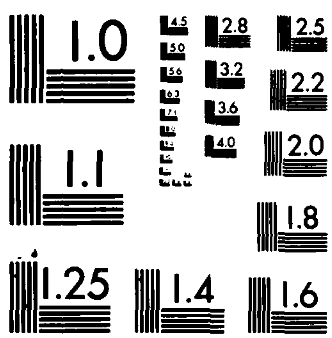
AD-A141 097 SPECIAL NONLINEAR OPTIMIZATION TECHNIQUES(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING A M RAGAB DEC 83 AFIT/MA/GOR/83D-5 1/2

UNCLASSIFIED F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A141 097

DTIC FILE COPY

SPECIAL NONLINEAR

OPTIMIZATION TECHNIQUES

THESIS

Ali M. Ragab
Colonel, Egypt

AFIT/MA/GOR/83D-5

DTIC
ELECTE
MAY 16 1984
A

This document has been approved for public release and sale; its distribution is unlimited.

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

84 05 15 013

SPECIAL NONLINEAR

OPTIMIZATION TECHNIQUES

THESIS

Ali M. Ragab
Colonel, Egypt

AFIT/MA/GOR/83D-5

Approved for public release; distribution unlimited

AFIT/MA/GOR/83D-5

SPECIAL NONLINEAR OPTIMIZATION TECHNIQUES

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Operations Research

Ali M. Ragab
Colonel, Egypt

December 1983

Approved for public release; distribution unlimited

Preface

My studies in mathematics and operations research are combined in this thesis to develop some techniques for non-linear optimization. Except where the theorems are specifically annotated, the work presented is my own.

This thesis would not have been possible without the guidance and interest of Dr. John Jones, Jr., whose unflagging optimism and cheerful help inspired and guided me through the successes and failures. I also wish to thank LtCol Peter Bobko and Dr.Dan W. Repperger for their professional comments. A word of thanks is also owed to the staff of the library of the School of Engineering for their efforts in making refrences available.

Ali M. Ragab

Table of Contents

List of Figures

List of Tables

Abstract

This thesis extends the work of Leighton and Jones which takes functions that satisfy special types of differential equations and determines an interval on which the functions either have zeros or attain bounded values. Theorems for locating zeros are proved for functions of a single variable and functions of several variables with illustrative examples. The applications of matrix equations to constrained optimization problems are described. An algorithm for random search technique for the general optimization problem is presented with a FORTRAN V program and test problems.

SPECIAL NONLINEAR

OPTIMIZATION TECHNIQUES

I. Introduction

For the past twenty years, considerable effort has been expended on the development of nonlinear programming theory and algorithms for solving nonlinear programming problems. Some of these algorithms have been implemented on digital computers. It is fair to say, however, that solving a complicated nonlinear programming problem by a computerized nonlinear programming algorithm is an automatic process. Unlike linear programming, where computerized algorithms(variations on the simplex method, usually) have long been able to solve problems of large size, nonlinear programming is still in its infancy as regards its ability to guarantee solutions to problems of even moderate size.

A major barrier for solving nonlinear programming problems is the lack of a computationally oriented way of representing nonlinear functions of n variables. The algorithms are often not as efficient as they could be because of the inability to compute automatically quantities related to the complicated nonlinear functional relationships that describe the models. For example, the accurate and speedy computation of first derivatives is a usual requirement for algorithms which solve system of nonlinear equations.

There is a basic dichotomy in programming algorithms: they may be designed to converge to local or global minima. A necessary condition for a point to be a local minimum of a differentiable function subject to constraints is due to Kuhn and Tucker (1) and might be considered the fundamental theorem of mathematical programming.

Assuming that the problem to be optimized is defined in some way, the various general methods of optimization can be conveniently classified as follows:

1. Analytical methods: which make use of the cla ical techniques of differential calculus and the calculus c ariations. These methods seek the extremum of a function f(X) by finding the values of X that cause the derivatives of f(X) with respect to X to vanish. When the extremum of f(X) is sought in the presence of constraints, techniques such as Lagrange multipliers and constrained variation are used. For the application of analytical methods, the problem to be optimized must be described in mathematical terms, so that the functions and variables can be manipulated by known rules. For large, highly nonlinear problems, analytical methods prove unsatisfactory.
2. Numerical methods: which use past information to generate better solutions to the optimization problem by means of iterative procedures.

A general summary of computer codes for mathematical programming that have been tested, documented, and are available to the public occurs in (2). Somewhat earlier there appeared a collection of FORTRAN listings of optimization codes, along with brief descriptions of the algorithms and their operations (3). The potential user is left to make his own choice as to which method will best serve his purpose.

This thesis investigates new methods in nonlinear optimization theory. The importance of this study is to be able to use these new methods in making decisions concerning modles for which classical techniques do not provide sufficient information. The overall objective is to address and resolve the new methods and apply them to some special problems.

Chapters 2 and 3 deal with objective functions of one variable or several variables. Application of given theorems provides a domain of good starting points for iterative methods.

Chapter 4 describes the applications of matrix equations to constrained optimization problems. Use is made of matrix equations to obtain solutions of certain classes of nonlinear equations.

Chapter 5 gives an algorithm for random search technique for the optimization problems of continuous variables or integer variables.

Chapter 6 gives conclusions and directions for further work in the areas covered by the thesis.

II. One Dimensional Unimodal Objective Functions

The main objective of this chapter is to obtain information as to the zeros, relative maxima, relative minima and bounds for one dimensional objective functions. The objective functions treated in this chapter are assumed differentiable and defined on an interval $[a,b]$. An objective function $f(x)$ will be required to be a solution of the differential equation of the form:

$$[A(x)f'(x)]'+C(x)f(x)=0 \qquad (2.1)$$

where

(i) $A(x)$ and $C(x)$ are both continuous functions on $[a,\infty]$ and $A(x) > 0$ on $[a,b]$,

(ii) $f'=df/dx$

Use will be made throughout this chapter of a class of trial functions defined by definition 2.1.

Definition 2.1

A trial function $u=u(x,h)$ is a real valued function of x and h, where h is a parameter, $h>0$ and u satisfies these conditions:

(1) $u > 0$

(2) $u(a,h) = u(b,h) = 0$ and

(3) u has at least one derivative with respect to x.

For example, on $[0,h]$, $h > 0$, one trial function (and the one used predominantly in this chapter) is of the form:

$$u = x(h-x)$$

Other trial functions which might be used on $[0,h]$ are

$$u = \sin(\pi x/h)$$

or

$$u = x^p (h-x)^q \qquad ,p,q \geq 1$$

Corresponding trial functions may be adapted to fit any interval $[a,b]$ or h,kh $,h>0$ and $K>1$, to extend the domain of these trial functions.

Associated with the given function f(x), the trial function u and the differential equation (2.1) will be a functional J(x,a,b) defined by

$$J(x,a,b) = -\int_b^a (Au_x^2 - C\,u^2)\,dx \qquad (2.2)$$

where,

$$u_x = du/dx$$

<u>Definition 2.2</u>

The functional J(x,a,b) above is a function defined on functions.

For example,the functional J(x,a,b) is a function defined on both f(x) and the trial function u.

Lighton (4) proved the following theorem to show that, whenever J(x,a,b) can be made negative by varying the parameter h, then at least one zero of f(x) must exist on the interval [0,h].

<u>Theorem 2.1</u>

If f(x) is a function which satisfies the linear differential equation (2.1), and if u is a trial function on [0,h] such that $J(x,0,h) < 0$ for the J(x,0,h) defined by(2.2) then f(x) has at least one zero on [0,h] .

The idea here is to make use of this theorem and the similar theorems in chapters 2,3 in optimization. If f(x) is the derivative of the objective function, then the zeros of f(x) correspond to the extreme points of the objective function. The range [0,h] for which J(x,0,h) is negative gives the range for these extreme points. For the unimodal functions it will be useless to search for the extreme points outside the range [0,h] . In most iterative methods which require a starting point, the range [0,h] is the recommended range of the starting point instead of searching over all the domain of the variable x. The choice of the starting point in [0,h] will decrease the computational time and the number of iterations. The following examples illustrate the application of theorem (2.1) for different functions that satisfy equation (2.1).

Example 2.1

The function y = sin(x) is known to be a solution of the differential equation

$$y'' + y = 0$$

A comparison with equation (2.1) gives

$A(x) = C(x) = 1$

then, with the trial function

$$u = x(h-x)$$

and varying h until J(u,0,h) becomes negative, i.e

$$J(u,0,h) = \int_0^h \{ (h-2x)^2 - (hx-x^2) \} \, dx < 0$$

solution of this inequality gives

$$h > \sqrt{10}$$

thus the function y has at least one zero on $(0, \sqrt{10})$.

NOTE: the actual zero occurs at $x = \pi = 3.1416$

Example 2.2

The function $f = P_n(x)$ is the Legendre polynomial which is a solution of the differential equation

$$(1-x^2)f'' - 2xf' + n(n+1)f = ((1-x^2)f')' + n(n+1)f = 0$$

Now, $A(x) = 1-x^2$ is positive for all $|x| < 1$,

$C(x) = n(n+1)$

choose

$$u = x(h-x) \text{ ,then}$$

$$J = \int_0^h \left\{ (1 - x^2)(h - 2x)^2 - (n^2+n)(xh - x^2)^2 \right\} dx < 0$$

the solution of this inequality gives

$$h > \sqrt{10/(n^2 + n + 4)} \qquad (2.3)$$

Table I compares the values of h calculated from equation (2.3) with the smallest positive zero of $P_n(x)$ calculated from equation (2.4) of the generalized Rodrigues formula for different values of n.

$$P_n(x) = \frac{1}{2^n n!} (d^{(n)}/dx^{(n)}) (x^2 - 1)^n \qquad (2.4)$$

TABLE I

Comparisons of h with zeros of $P_n(x)$

n	h	zeros of $P_n(x)$
3	.7906	.7746
4	.6455	.3400
5	.5423	.5385

Example 2.3 Mathews (5)

Let y be the objective function given by the solution of the ordinary differential equation (2.5)

$$y'' + xy = 0 \qquad (2.5)$$

In this example we are to approximate the first zero of Airy's function y(x) shown in Figure 1. Matching of equations (2.5) and(2.1) gives

$$A(x) = 1 \quad , \quad C(x) = x$$

Let $u = x(h-x)$

$$J(u,0,h) = \int_0^h \left\{ (h-2x)^2 - x(hx - x^2)^2 \right\} dx < 0$$

The solution of this inequality gives

$$h > (20)^{1/3}$$

then there is at least one zero for y on $[0,(20)^{1/3}]$.

Komkov(6) developed theorem (2.2) that is a generalization of theorem (2.1). In the general case we will consider

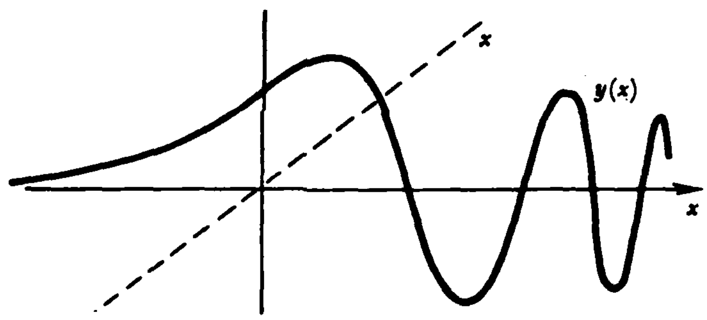


Figure 1. Airy's Function

the nonlinear equation

$$[\, A(x)V' \,]' + C(x)f(V) = 0 \qquad (2.6)$$

where

(i) $x \in [x_0, \infty)$,

(ii) $A(x) > 0$ for all $x \in [a,b]$ and $A(x) \in C^2 \; [x_0, \infty)$,

(iii) $C(x) \in C \; [x_0, \infty)$, and

(iv) $f(V)$ is of the form $f(V) = V^k$, k is a positive integer and $k > 1$.

<u>Theorem 2.2</u>

Let u be a $C^2 \; [x_0, \infty)$ trial function, and $G(u)$ be $C^1(-\infty, +\infty)$ such that $G\{u(a)\} = G\{u(b)\} = 0$, $G\{u(X)\} > 0$ for all $a < x < b$. Let $g(u)$ denote $g(u) = G'(u)$, and let $g^2(u)/G(u)$ be a bounded function of x on the interval $[a,b]$, and m denotes

$$m = \max_{x \in [a,b]} \left(\frac{g^2(u)}{4G(u)} \right) \qquad (2.7)$$

let

$$J(u,a,b) = \int_a^b (Au'^2 - CG)\, dx < 0 \qquad (2.8)$$

then any solution $V(x)$ of equation (2.6) , with $f(V) = V^k$, $k > 1$ which satisfies $V(a) \geq 0$ will satisfy the inequality

$$|V(x)| < \left(\frac{m}{k}\right)^{1/(k-1)}$$

on some subinterval of $[\, a,b \,]$.

The following examples illustrate the applications of theorem (2.2) for different objective functions.

Example 2.4 -Komkov(5)

The Emden's equation (2.9) occurs in astrophysics. It arises in the discussion of a simplified thermodynamic model of a contracting nebular cloud.

$$y'' + (2/x)y' + y^n = 0 \qquad (2.9)$$

When equation (2.9) is reduced to the form of equation (2.6) it gives

$$(x^2 y')' + x^2 y^n = 0$$

In the special case which occurs in physics, n=5, the following solution is known

$$y = \left(\frac{3c}{x^2 + c^2}\right)^{1/2} \qquad (2.10)$$

where,c is an arbitrary nonnegative constant.

A complete solution to the Emden's equation is not known.

Application of theorem (2.2) allows us to estimate how close the solutions approach zero on a given interval. Let the objective function y be given by equation (2.10). This function satisfies the differential equation (2.6) with

$$A(x) = x^2 \qquad \text{and} \qquad C(x) = x^2$$

choose $G(u) = 4u^2$ and $u = x(h-x)$

The technique is to use equation (2.8) to find the smallest positive h for which J is negative, i.e

$$0 > J = \int_0^h (Au'^2 - C\,G)\,dx$$

$$= \int_0^h \{ x^2(h-2x)^2 - 4x^2(h-x) \} \quad dx$$

The solution of this inequality gives

$$h > 1.8708$$

Komkov (6) solved the problem for h = 2 and found that m = 4. Hence every solution V(x) of equation (2.9) with n = 5 will attain a value

$$|V(x)| < (4/5)^{1/4} \cong .945$$

on some subinterval of [0,2] .

<u>Example 2.5</u> -Komkov(6)

Consider the equation

$$y'' + (n + 1/2 - x^2/4)\,y = 0 \qquad (2.11)$$

with n = 6.

A solution of this equation is given by the parabolic cylinder function $y_6(x)$, which may be expressed in terms of the Hermite polynomial $H_6(x)$

$$y_6(x) = \frac{e^{-(x^2/4)}}{\sqrt{6!/\sqrt{2\pi}}} \quad H_6(x) \qquad (2.12)$$

where

$$H_6(x) = x^6 - 15x^4 + 45x^2 - 15$$

$y_6(x)$ vanishes at $x \cong .62$ and at $x \cong 1.90$

Komkov (5) proved that every solution of equation (2.11) vanishes on $[0,1.8]$. He used fixed $h = 1.8$. If we use h as a parameter and let

$$u = \pi x/h$$

choose $G(u) = \sin^2 u$

then $g(u) = 2\sin u.\cos u$

Hence

$$g^2(u)/4 = \sin^2 u.\cos^2 u \le G(u) = \sin^2 u$$

then theorem (2.2) can be applied. Comparison of equation (2.11) with equation (2.6) gives

$$A(x) = 1,$$

$$C(x) = n + 1/2 - x^2/4$$

$$0 > J(u,0,h) = \int_0^h (\; Au'^2 - CG \;)\; dx$$

$$= \int_0^h \left\{ (\; \pi/h)^2 - (13/2 - x^2/4)\sin^2(\pi x/h)\right\} dx$$

solution of this inequality gives

$$\pi^2/h - 13h/4 - h^3/16\pi < 0$$

using some experimental values of h we have

$$J(u,0,1.77) = 0.01941 \quad ,$$

$$J(u,0,1.78) = -0.041074$$

Thus, choosing $h = 1.78$ indicates that the function y has at least one zero on $[0,1.78]$ which improves on Komkov's result. The improvement in the result is due to varying h as a parameter instead of choosing constant value for $h = 1.8$.

Theorem 2.3

Let the objective function V(x), or it's derivative be a solution of the ordinary differential equation (2.13) on $x \in [a,b]$

$$\{ A(x).D(V).V' \}' + 2B(x).D(V).V' + C(x).f(V) = 0 \qquad (2.13)$$

where,(') indicates differentiation with respect to x.

If the following conditions are true :

(1) the function A is continuously differentiable with respect to x and $A > 0$ on $[a,b]$,

(2) the functions D and f are continuously differentiable with respect to V,

(3) the functions B and C are continuous on [a,b],

(4) $f = 0$ only if $V = 0$,

(5) Q(x) is positive definite for some continuous function E(x) on [a,b] where

$$Q(x) = \begin{bmatrix} A & -B \\ -B & E \end{bmatrix} \qquad (2.14)$$

(6) $J(u,a,b) < 0$ with

$$J(u,a,b) = \int_a^b \left\{ Au'^2 - 2Buu' + (E - C)u^2 \right\} dx$$

then, either

(i) f and V have zeros on [a,b] , or

(ii) $D^2 > D(df/dV)$ on some subinterval $p,q \subseteq [a,b]$.

Proof:

Let u(x) be given as a trial function with $u(b) = 0$ and $u(a) = 0$. If V has a zero on [a,b]then the conclusion follows immediately.

Suppose that $V \neq 0$ on $[a,b]$ and hence $f \neq 0$ also on $[a,b]$. Since $u(a) = u(b) = 0$

then

$$\int_a^b A.D.V'.u^2(1/f) \quad dx = 0 \tag{2.15}$$

Also, since Q is positive definite, then, by equation (2.14) it follows that

$$0 < \begin{bmatrix} u'-uDV'(1/f) & , u \end{bmatrix} \begin{bmatrix} A & -B \\ -B & E \end{bmatrix} \begin{bmatrix} u'-uDV'(1/f) \\ u \end{bmatrix} \tag{2.16}$$

From equation (2.15) and equation (2.16) the following inequality holds:

$$0 < \int_a^b \left\{ A(u' - \frac{uDV'}{f})^2 - 2uB(u' - \frac{uDV'}{f}) + Eu^2 \right.$$
$$\left. + (ADV'u^2/f) \right\} dx$$

$$= \int_a^b \left\{ Au'^2 - 2Au'V'D(1/f) + Au^2D^2V'^2(1/f^2) - 2Buu' \right.$$
$$+ 2Bu^2DV'(1/f) + Eu^2 + (ADV')'u^2(1/f)$$
$$\left. + ADV'(u^2/f)' \right\} dx$$

$$= \int_a^b \left\{ Au'^2 - 2Buu' + Eu^2 - 2Auu'V'D(1/f) \right.$$
$$+ Au^2D^2V'^2(1/f^2) + 2Bu^2DV'(1/f) - 2BDV'u^2(1/f)$$
$$\left. +2ADV'uu'(1/f) - ADV'u^2V'(df/dV)(1/f^2) - Cfu^2(1/f) \right\} dx$$

$$= \int_a^b \left\{ Au'^2 - 2Buu' + (E - C)u^2 \right\} dx$$

$$+ \int_a^b Au^2V'^2(1/f^2)(D^2 - D\frac{df}{dV}) \quad dx$$

$$= J(u,a,b) + \int_a^b Au^2V'^2(1/f^2)(D^2 - D\frac{df}{dV}) \quad dx$$

Since $J(u,a,b) < 0$

then, it follows that either

(i) $V = 0$ and $f = 0$, or

(ii) $D^2 > D\frac{df}{dV}$ on some subinterval $p,q \subsetneq [a,b]$.

Example 2.6 -Mahfoud (7)

Let $V(x) = (\sin x)^{1/3}$ be a given function that satisfies the ordinary differential equation

$$(V^2 V')' + V^3/3 = 0 \qquad (2.17)$$

A comparison of equation (2.17) with equation (2.13) gives

$A = 1$, $D = V^2$, $C = 1/3$, $f = V^3$ and $B = 0$

Let $u = x(h-x)$ be the trial function for $h > 0$, then

$$0 > J(u,0,h) = \int_0^h (Au'^2 - Cu^2) \, dx$$

$$= \int_0^h \left\{ (h - 2x)^2 - (1/3)(hx - x^2) \right\} dx$$

Solution of this inequality gives

$$h > (30)^{1/2}$$

then, either

(i) V has at least one zero on $[0,(30)^{1/2}]$, or

(ii) $D^2 = V^4 > V^2 (df/dV) = V^2 . 3V^2 = 3V^4$ which is impossible.

Then, V has at least one zero on $[0,(30)^{1/2}]$.

Example 2.7

Let V(x) be an objective function which satisfies the ordinary differential equation

$$(V^2V')' + (k/x)V^{2n+1} = 0 \qquad (2.18)$$

where n is a positive integer and k is a real constant.

A comparison of equation (2.13) with equation (2.18) gives

$$A = 1\ , \quad D = V^{2n}\ , \quad C = k/x\ , \quad \text{and} \quad f = V^{2n+1}$$

Let $u = x(h - x)$ be the trial function. Then

$$0 > J(u,0,h) = \int_0^h (Au'^2 - Cu^2)\,dx$$

$$= \int_0^h \left\{ (h - 2x) - (k/x)(hx - x^2)^2 \right\} dx$$

Solution of this inequality gives

$$h > 4/k$$

Then, either

(i) V(x) has at least one zero on $[0,4/k]$, or

(ii) $D^2 = V^{4n} > D(df/dV) = V^{4n}(2n+1)$ which is impossible for positive values of n.

Then, V(x) has at least one zero on $[0,4/k]$.

Example 2.8

Consider the Čebyšev polynomial function which is a solution to the Čebyšev ordinary differential equation

$$\{(1 - x^2)y'\}' + xy' + n^2 y = 0$$

where, n is a positive integer.

Here we have

$$A = 1 - x^2, \quad D = 1, \quad B = x/2 \quad \text{and} \quad C = n^2$$

Let $u = x(h - x)$ be the trial function and choose

$$E(x) = \text{constant}$$

such that

$$Q = \begin{bmatrix} A & -B \\ -B & E \end{bmatrix} = \begin{bmatrix} 1-x^2 & -x/2 \\ -x/2 & E \end{bmatrix}$$

is positive definite.

Assume $n = 2$, then to find $h > 0$ such that y has at least one zero on $[0,h]$, compute

$$0 > J(u,0,h) = \int_0^h \{ Au'^2 - 2B\,uu' + (E - C)u^2 \}\, dx$$

$$= \int_0^h \{ (1 - x^2)(h - 2x)^2 - 2(x/2)(h - 2x)(hx-x^2) + (E - 4)(hx - x^2)^2 \}\, dx$$

Solution of this inequality gives

$$h > \frac{2(5)^{1/2}}{(15 - 2E)^{1/2}} \qquad (2.19)$$

then, for a suitable choice of $E < 15/2$ that makes $Q(x)$ positive definite, we have either,

(i) $y(x)$ has at least one zero on $[0,h]$, or

(ii) $D^2 = 1 > D(df/dV) = 1$ which is impossible.

Thus y has at least one zero on $[0,h]$, h is given by (2.19).

Algorithm for Numerical Solution

This algorithm determines the extreme point of a unimodal function (the zero of the derivative) numerically.

The inputs of the algorithm are:

1- The derivative of the objective function.

2- The limits of the variable $x \in (a,b)$.

3- The accuracy required to determine the extreme point.

The output of the algorithm is the value of h that makes the functional J negative. The value of J is calculated by the Romberg integration algorithm.

Step 1: Initialization.

Step 2: Choose initial value $h = a + d$ where d is the step size for the increment in h.

Step 3: Calculate J by Romberg algorithm for the limits a and h.

Step 4: If J is negative go to step 8.

Step 5: If $J \geq 0$ put $h = h + d$.

Step 6: If $h \geq b$ go to step 10.

Step 7: Go to step 3.

Step 8: Print h and h-d.

Step 9: Go to step 11.

Step 10: Print "No zero between a,b.

Step 11: Stop.

The zero of the derivative of the objective function lies between h-d and h. The accuracy of determining the zero depends on d. The computer program of the above algorithm is written in FORTRAN V and is attached in appendix A.

The numerical solution of Example 2.3 by the algorithm is:

$$2.700 \leq h \leq 2.701$$

for d = .001

The case of multimodal functions of one variable is just an extension of the case of unimodal function. The idea is the transformation of the origin to the first h that makes J negative. Following the same steps as in the case of unimodal functions but with the new origin at h the algorithm can locate the second extreme point. Other extreme points can be located by further transformations of the origin.

III. Special Objective Functions of Several Variables

In this chapter objective function $W(X) = W(x1,x2,..,xn)$ will be considered where the variables $x1,x2,..,xn$ are bounded. The notion of an n-dimensional trial function $u(x1,x2,..,xn,h1,h2,..,hn)$ will be used and is an extension of the $u(x,h)$ used in chapter 2. The theorems are presented for the general case of n variables. Proofs will be given only for the case of two variables for simplicity (see Figure 2); however, they can be extended to apply to functions of several variables.

3.1 Objective Functions that Satisfy First Order Partial Differential Equations

The objective function $W(X)$ is required to be a solution of the first order differential equation

$$\sum_{i=1}^{n} [\, Ai(X).B(W).W(X)]_{xi} + C(X).f[W(X)] = 0 \qquad (3.1)$$

where

$X = (x1,x2,..,xn)$, withen the region R defined by

$$R = \left\{ (x1,x2,..,xn) : 0 \le x1 \le a1,\ 0 \le x2 \le a2, ..., 0 \le xn \le an \right\}, \qquad (3.2)$$

$[A]_{xi}$ denotes the derivative of A with respect to xi

Theorem 3.1

Let $W(X)$ be a given objective function which satisfies the differential equation (3.1) where

(1) $Ai(X) > 0$ and continuously differentiable over R defined by (3.2), $i = 1,2,...,n$.

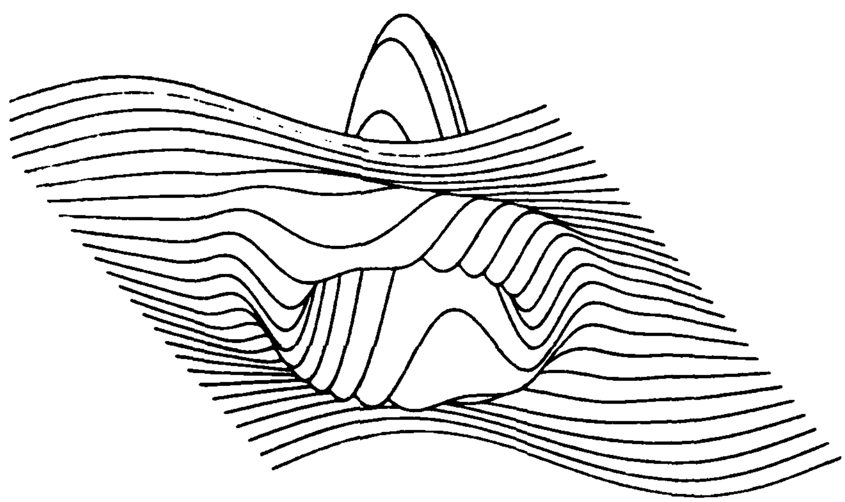


Figure 2. A Function of Two Variables

(2) $B(W)$ is continuously differentiable function of W.

(3) $f[W(X)] = 0$ whenever $W(X) = 0$.

(4) $C(X)$ is continuous on R.

Let

$$J = \int_0^{h1} \int_0^{h2} \cdots \int_0^{hn} \left[\sum_{i=1}^{n} Ai.u_{xi}^2 - C.G(u) \right] dX \leq 0 \qquad (3.3)$$

where

(i) $u = u(x1,x2,\ldots,xn,h1,h2,\ldots,hn)$ is a trial function differentiable on R and is zero on the boundry of R, and positive in the interior of R.

(ii) $G(u) > 0$ in the interior of R and vanishes on the boundary of R.

(iii) $dX = dx1\ dx2 \ldots .dxn$.

then, either

(a) $W(X)$ has at least one zero on H, where

$$H = \left\{ (x1,x2,\ldots,xn) : 0 \leq x1 \leq h1,\ 0 \leq x2 \leq h2, \ldots, 0 \leq xn \leq hn \right\} \qquad (3.4)$$

or

(b) $$\sum_{i=1}^{n} Ai[(1/4)g^2(u)B^2W^2 - BG(u)W_{xi}(df/dW] > 0 \qquad (3.5)$$

on some subregion $H \subseteq R$, where

$$g(u) = dG(u)/du$$

Proof:

To prove theorem (3.1) we will use a special case for $n = 2$ and the proof can be extended for the case of n-variables. Assume $W(X)$ does not vanish on R and consequently $f[W(X)]$ will not. For simplicity weshall use A,u,g,B,f,W and G without arguments. Also derivatives will be represented by subscripts ($du/dx1 = u_{x1}$).

By Green's Divergence theorem (8) we have

$$\int_0^{h1}\int_0^{h2} [(A1.B.W.G/f)_{x1} + (A2.B.W.G/f)_{x2}]\, dx2\, dx1 = 0$$

then, the following inequality holds:

$$0 < \int_0^{h2}\int_0^{h1} [\, A1(u_{x1} - gBW/2f)^2 + A2(u_{x2} - gBW/2f)^2$$

$$+ (A1BWG/f)_{x1} + (A2BWG/f)_{x2} \,]\, dx1\, dx2$$

$$= \int_0^{h2}\int_0^{h1} [A1.u_{x1}^2 - A1.gBWu_{x1}.1/f + A1.g^2.B^2.W^2(1/4f^2)$$

$$+ A2.u_{x2}^2 - A2.g.B.W.u_{x2}(1/f) + A2.g^2.B^2.W^2(1/4f^2)$$

$$+ (A1.B.W)(f.g.u_{x1} - G.W_{x1}.df/dW)(1/f^2) + (A1.B.W)_{x1}(G/f)$$

$$+ (A2.B.W)(f.g.u_{x2} - G.W_{x2}.df/dW)(1/f^2)$$

$$+ (A2.B.W)_{x2}(G/f) \,]\, dx1\, dx2$$

$$= \int_0^{h2}\int_0^{h1} [A1.u_{x1}^2 + A2.u_{x2}^2 - C.G] + [\sum_{i=1}^{2}(Ai/f^2)(g^2B^2W^2/4$$

$$- B.W.G.W_{xi}.df/dW)]\, dx1\, dx2$$

$$=J(u,h1,h2) + \int_0^{h2}\int_0^{h1} [\sum_{i=1}^{2} (Ai/f^2)(g^2.B^2.W^2.1/4$$

$$- B.W.G.W_{xi}.df/dW)]\ dx1\ dx2$$

Since $J(u,h2,h1) < 0$ by hypothesis, then, either

(a) $W(X)$ has at least one zero on H, or

(b) $\sum_{i=1}^{2} Ai[g^2.B^2.\ W^2/4 - B.G.W_{xi}.df/dW\ ] > 0$

on some region $H \subsetneq R$

Theorem (3.1) says that: if a region $H \subsetneq R$ can be found for which a trial function u makes $J(u,h1,h2) < 0$, then, W will either vanishes somewhere on R , or, the inequality (3.5) is satisfied on some region of R.

3.2 Objective Functions that Satisfy Elliptic Partial Differential Equation

Theorem (3.1) applies to certain functions that satisfy the first order partial equation (3.1). For other class of functions that satisfy the elliptic partial differential equation we have to use the following theorem:

Theorem 3.2

Let the objective function V(X) satisfy the elliptic partial differential equation

$$\sum_{i,j=1}^{n} [a_{ij}(X).B(V).V_{xi}\ ]_{xj} + b(X).f[V(X)] = 0 \qquad (3.6)$$

where

(1) $a_{ij}(X)$ is an element of a symmetric positive definite matrix A, and has continuous partial derivatives with respect

to X.

(2) $X = (x1, x2, x3, \ldots, xn)$ is a point belonging to a smooth closed bounded region R; R is defined by equation (3.2).

(3) B(V) and f(V) are nonlinear functions of V(X) such that $f(V) \neq 0$ when $V \neq 0$.

Let the functional J(u) associated with V(X) and equation (3.6) be of the form

$$J(u) = \int_0^{hn} \cdots \int_0^{h2} \int_0^{h1} \Big[\sum_{i,j}^{n} a_{ij}(X) u_{xi} u_{xj} - b(X) u^2 \Big] \, dX$$

where

(i) $dX = dx1\, dx2 \ldots\ldots dxn$

(ii) the trial function u(X) has continuous partial derivatives with respect to $x1, x2, \ldots, xn$ and it is also zero on the boundary of R and positive in the interior of R.

then, for $J(u) < 0$, either

(a) V(X) has at least one zero on some region H = R, and H is defined by equation (3.4), or

(b) $B^2/f^2(V) > B(df/dV)$ for $H \subseteq R$.

Proof:

We will use the same methodology as in proof of theorem (2.1) with the same simplicity assumptions for n = 2.

Assume V(X) does not vanish on R, and consequently f[V(X)] will not. Since A is a positive definite matrix, let

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

then

$$(u_{x1} - uBV_{x1}.1/f\ ,\ u_{x2} - uBV_{x1}.1/f)A\begin{bmatrix} u_{x1} - uBV_{x1}.1/f \\ u_{x2} - uBV_{x2}.1/f \end{bmatrix} > 0 \tag{3.7}$$

Also by Green's theorem (8) we have

$$\int_0^{h2}\int_0^{h1}\Big\{[u^2(a_{11}BV_{x1} + a_{12}BV_{x2}).1/f]_{x1}$$

$$+ [u^2(a_{21}BV_{x1} + a_{22}BV_{x2}).1/f]_{x2}\Big\}dx1\ dx2 = 0 \tag{3.8}$$

Combination of equations (3.7) and (3.8) gives the following inequality:

$$0 < I = \int_0^{h2}\int_0^{h1}\Big\{a_{11}u_{x1}^2 - 2a_{11}uBu_{x1}V_{x1}.1/f + a_{11}u^2B^2V_{x1}^2.1/f^2$$

$$+a_{12}u_{x1}u_{x2} - a_{12}uBu_{x2}V_{x1}.1/f - a_{12}uBu_{x1}V_{x2}.1/f$$

$$+ a_{12}u^2B^2V_{x1}V_{x2}.1/f^2 + a_{21}u_{x2}u_{x1} - a_{21}uBu_{x2}V_{x1}.1/f$$

$$- a_{21}uBu_{x1}V_{x2}.1/f + a_{21}u^2B^2V_{x1}V_{x2}.1/f^2 + a_{22}u_{x2}^2$$

$$- 2a_{22}uBu_{x2}V_{x2}.1/f + a_{22}u^2B^2V_{x2}^2.1/f^2$$

$$+ [u^2(a_{11}BV_{x1} + a_{12}BV_{x2}).1/f\ ]_{x1}$$

$$+ [u^2(a_{21}BV_{x1} + a_{22}BV_{x2}).1/f\ ]_{x2}\Big\}\ dx1\ dx2$$

$$= \int_0^{h2}\int_0^{h1}\Big\{\ (\sum_{i,j=1}^{2}a_{ij}u_{xi}u_{xj}) + (-2a_{11}uBu_{x1}V_{x1}.1/f$$

$$+ a_{11}u^2B^2V_{x1}^2 - a_{12}uBu_{x2}V_{x1}.1/f - a_{12}uBu_{x1}V_{x2}.1/f$$

$$+ a_{12}u^2B^2V_{x1}V_{x2}.1/f^2 - a_{21}uBu_{x2}V_{x1}.1/f - a_{21}uBu_{x1}V_{x2}.1/f$$

$$+ a_{21}u^2B^2V_{x1}V_{x2}.1/f^2 - 2a_{22}uBu_{x2}V_{x2}.1/f$$

$$+ a_{22}u^2B^2V_{x2}^2.1/f^2 \;)$$

$$+ (a_{11}BV_{x1} + a_{12}BV_{x2})(2fuu_{x1} - u^2.df/dV.V_{x1})(1/f^2)$$

$$+ (a_{21}BV_{x1} + a_{22}BV_{x2})(2fuu_{x2} - u^2.df/dV.V_{x2})(1/f^2)$$

$$+ (u^2/f)[(a_{11}BV_{x1} + a_{12}BV_{x2})_{x1} + (a_{21}BV_{x1} + a_{22}BV_{x2})_{x2}]\Big\}dx1dx2$$

and by equation (3.6) we have

$$0 > I = \int_0^{h2}\int_0^{h1} [(\sum_{i,j=1}^{2} a_{ij}u_{xi}u_{xj}) - bu^2$$

$$+ \sum_{i,j=1}^{2} (a_{ij}u^2V_{xi}V_{x_j}.1/f^2)(B^2 - B.df/dV)]\; dx1\; dx2$$

$$= J(u,h1,h2)$$

$$+ \int_0^{h2}\int_0^{h1} \sum_{i,j=1}^{2} a_{ij}u^2V_{xi}V_{xj}(1/f^2)(B^2 - B.df/dV)\; dx1\; dx2$$

Since $\sum_{i,j=1}^{2} a_{ij}V_{xi}V_{xj}$ is positive because A is positive definite, and since $J < 0$ by hypothesis, it follows that either

(i) V has at least one zero on R, or

(ii) $B^2/f^2 > B.\ df/dV$ on a subregion $H \subseteq R$.

The following example illustrates how to apply that theorem for a practical function.

<u>Example 3.1</u>

Let V(x,y) be a function which satisfies the differential equation:

$$V_{xx} + V_{yy} + k_1 V + k_2 V^3 = 0 \tag{3.9}$$

choose the trial function:

$$u = \sin(\pi x/h_1) \sin(\pi y/h_2)$$

a comparison of equations (3.9) and (3.6) gives

$$a_{11} = a_{22} = 1 \quad \text{and}$$
$$a_{12} = a_{21} = 0$$

which satisfies that A is a positive definite matrix. Also, we have

$$b = 1, \quad B = 1 \quad \text{and}$$
$$f = k_1 V + k_2 V^3$$

$$J = \int_0^{h_2} \int_0^{h_1} (u_x^2 + u_y^2 - u^2)\, dx\, dy$$

$$= \int_0^{h_2} \int_0^{h_1} \Big\{ [(\pi/h_1) \cos(\pi x/h_1) \sin(\pi y/h_2)]^2$$

$$+ [(\pi/h_2) \sin(\pi x/h_1) \cos(\pi y/h_2)]^2$$

$$- [\sin(\pi x/h_1) \sin(\pi y/h_2)]^2 \Big\}\, dx\, dy.$$

calculatin of the double integration gives

$$J = (h_1 h_2/4)(\pi^2/h_1^2 + \pi^2/h_2^2 - 1)$$

Since $J < 0$ and h_1, h_2 are positive, then

$$(\pi^2/h_1^2) + (\pi^2/h_2^2) < 1 \qquad (3.10)$$

By theorem (3.2), either

(a) $V(x,y)$ has at least one zero on $H \subseteq R$ defined by

$H = \left\{ (x,y) : 0 \leq x \leq h_1,\ 0 \leq y \leq h_2 \right\}$, where, h_1 and h_2 satisfy the inequality (3.10), or

(b) $B^2/f^2 > B.df/dV$ which gives

$$1/(k_1V + k_2V^3)^2 > k_1 + 3k_2V^2$$

i.e.

$$V^2(k_1 + k_2V^2)^2\ (k_1 + 3k_2V^2) < 1 \qquad (3.11)$$

for $k_1 = 0$, the inequality (3.11) gives

$$3k_2^2V^8 < 1$$

which gives an upper bound of V

$$V < (3k_2^2\ )^{-1/8}$$

for $k_2 = 0$, the inequality (3.11) gives

$$V^2k_1^3 < 1$$

which gives an upper bound of V

$$V < (k_1^3\ )^{-1/2}$$

If for some values of k_1 and k_2 the inequality (3.11) does not hold, this makes case (a) true and there will be at least one zero on H.

Algorithm for Numerical Solution

This algorithm finds values of h_1 and h_2 that make the functional J negative. Since J is a double integral (for the case of two variables), that can be defined by

$$J = \int_A^{h_2} \int_C^{h_1} f(x,y)\, dy\, dx$$

then the idea is to find the values of the parameters h_1 and h_2 that make J negative.

Step 1: Initialization.

Step 2: Input: lower bounds A and C and upper bounds U_1 and U_2 for x and y respectively; initial values h_1 and h_2; incremental values d_1 and d_2 for x and y respectively.

Step 3: Find the approximation of J by any multiple integration algorithm for the lower bounds A and C and the upper limits h_1 and h_2.

Step 4: If $J \geq 0$, then let $h_1 = h_1 + d_1$ and $h_2 = h_2 + d_2$, otherwise, go to Step 6.

Step 5: If $h_1 \leq U_1$ and $h_2 \leq U_2$, go to Step 3, otherwise, go to Step 7.

Step 6: If $J < 0$, print h_1 and h_2.

Step 7: Stop.

If the value of J will be negative for $x = h_1$ and $y = h_2$, then, the extreme point of the function will be in the range

$$h_1 - d_1 \leq x \leq h_1 \quad \text{and}$$
$$h_2 - d_2 \leq y \leq h_2$$

The algorithm is written in FORTRAN V and is attached in Appendix B. The double integral is approximated using the

Composite Simpson's Method.

For the case of multimodal functions, the idea is just the transformation of the axes to the point $(x,y) = (h_1,h_2)$, where (h_1,h_2) is the point determined by the previous routine. The first transformation is used to locate the second extreme point, and so on.

Figure 3 shows the first extreme point in region A. The origin is shifted to $O_1(H_1,H_2)$. The second extreme point is located in region B.

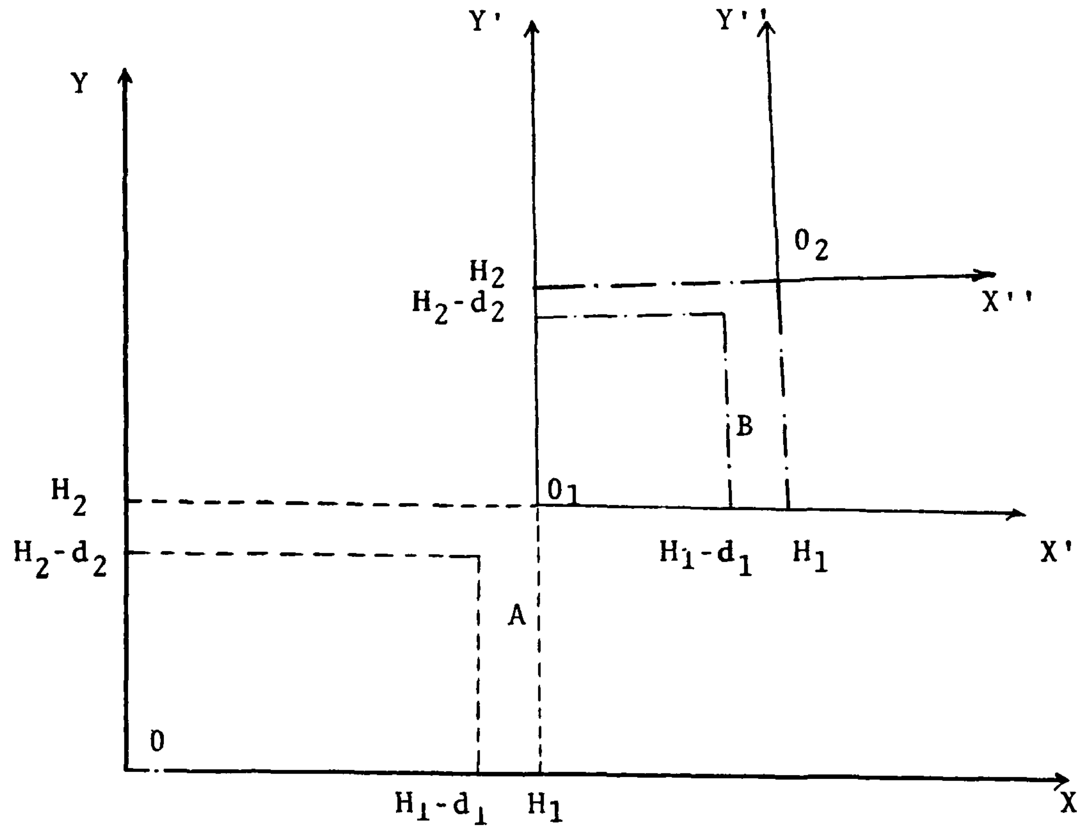


Figure 3. Location of Extreme Points

3.3 Use of the Generalized Inverse Technique in Optimization

Now we consider a general matrix A of order mxn and rank k which may be less than min(m,n) and raise the question whether an inverse exists in some suitable sense. This naturally depends on the purpose for which such an inverse is used.

If A is a nonsingular matrix of order m, the solution of the linear equation

$$AX = B$$

where B is a mx1 column vector, is given by

$$X = A^{-1}B$$

where A^{-1} is the inverse of A (i.e., $AA^{-1} = I$). We ask the question whether a similar representation of the solution, that is , of the form

$$X = GB$$

is possible when A is a singular square or rectangular matrix. If there exists a matrix G such that X = GB is a solution of AX = B for any B such that AX = B is a consistent equation, then G does the same job (or behaves) as the inverse of A, hence may be called a generalized inverse of A.

<u>Definition 3.1</u>

Let A be an mxn matrix of arbitrary rank. A generalized inv. of A is an nxm matrix G such that X = is a solution of the equation AX = B for any B which makes the equation consistent.

<u>Lemma 3.1</u>

G exists iff

$$AGA = A \tag{3.12}$$

Rao and Mitra (9) proved the following theorm:

Theorem 3.3

Let A be of order mxn and G be any generalized inverse of A. Further let H = GA. Then the following hold;

(a) A general solution of the homogeneous equation AX = 0 is

$$X = (I - H)Z$$

where Z is an arbitrary vector.

(b) A general solution of a consistent nonhomogeneous equation

$$AX = B \tag{3.13}$$

is

$$X = GB + (I - H)Z \tag{3.14}$$

(c) A necessary and sufficient condition that AX = B is consistent is that

$$AGB = B \tag{3.15}$$

3.3.1 Common Zeros of Nonlinear Functions

The idea here is to use the generalized inverse technique in constrained optimization problems. The following two examples illustrate this technique.

Example 3.2

Given the nonlinear function

$$F(x,y) = xy^2 + y^3 - x^2 = 0 \tag{3.16}$$

with the constraint

$$V(x,y) = x + y - 1 = 0 \tag{3.17}$$

the problem is to find common zeros of both F(x,y) and V(x,y)

Writing equation (3.17) in the matrix form given by (2.13) gives

$$V(x,y) = (1\ ,\ 1)\begin{bmatrix} x \\ y \end{bmatrix} = (1) \qquad (3.18)$$

where,

$$A = (1\ ,\ 1), \quad B = (1) \quad \text{and } X = \begin{bmatrix} x \\ y \end{bmatrix}$$

The generalized inverse of A (see(10) for calculation of G) is

$$G = \begin{bmatrix} .5 \\ .5 \end{bmatrix}$$

This can be checked by applying lemma (3.1) as follows:

$$AGA = (1\ ,\ 1)\begin{bmatrix} .5 \\ .5 \end{bmatrix}(1\ ,\ 1) = (1\ ,\ 1) = A$$

Applying the consistency condition (3.15) gives

$$AGB = (1\ ,\ 1)\begin{bmatrix} .5 \\ .5 \end{bmatrix}(1) = (1) = B$$

Then by theorem (3.3), the general solution of equation(3.14) is

$$X = GB + (I - H)Z$$

where

$$H = GA = \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix}$$

then, if we assume

$$Z = \begin{bmatrix} u \\ v \end{bmatrix}$$

we have

$$X = \begin{bmatrix} .5 \\ .5 \end{bmatrix} (1) + \begin{bmatrix} .5 & -.5 \\ -.5 & .5 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$= \begin{bmatrix} .5 + .5u - .5v \\ .5 - .5u + .5v \end{bmatrix} \tag{3.19}$$

A Taylor series expansion of F(x,y) gives

$$\begin{aligned} F(x,y) = {} & F(0,0) + [x.F_x(0,0) + y.F_y(0,0)] \\ & +(1/2!)[x^2.F_{xx}(0,0) + 2xy.F_{xy}(0,0) + F_{yy}(0,0)\ ] \\ & +(1/3!)[x^3.F_{xxx}(0,0) + 3x^2y.F_{xxy}(0,0) + 3xy^2.F_{xyy}(0,0) \\ & \qquad + y^3.F_{yyy}(0,0)\ ] + \ldots\ldots. \end{aligned}$$

$$\begin{aligned} & = (1/2)(x\ ,\ y) \begin{bmatrix} -2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ & + (1/6)(x\ ,\ y) \begin{bmatrix} 0 & 6y \\ 0 & 6y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned} \tag{3.20}$$

Substituting in equation (3.20) by the general solution given by equation (3.19) gives

$$F(x,y) = -(.5 + .5u - .5v)^2 + y(.5 - .5u + .5v) = 0 \tag{3.21}$$

Since the general solution given by equation (3.14) is valid for all values of Z, [theorem (3.3)(b)], then choosing

$$Z = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and substituting in equation (3.21) gives

$$-.25 + .5y = 0 \qquad \text{or,}$$

$$y = .5$$

Using the constraint (3.17) gives

$$x = .5$$

Then the common zero of equations (3.16) and (3.17) is

$(x,y) = (.5,.5)$

The next example illustrates the same technique but for equations of four variables.

Example 3.3

Find the common zeros of the pair of nonlinear functions:

$$F(x,y,u,v) = x^2 - 2y^2 - u^2 + w^2 = 4 \qquad (3.22)$$

$$V(x,y,u,v) = x - 2y + u + w = 8 \qquad (3.23)$$

Writing equation (3.23) in the matrix form given by equation (3.13) gives

$$V = (1,-2,1,1)(x,y,u,w)^T = (8)$$

where T denotes the transpose of the matrix.

Assume we know the generalized inverse G of A to be

$$G = (\ 3/8,\ -1/8,\ 2/8,\ 1/8\ )^T \qquad (3.24)$$

which satisfies lemma (3.1) and the consistency condition. By theorem (3.3), the general solution of equation (3.23) is

$$X = GB + (I - H)Z$$

where

$$H = GA = \begin{bmatrix} 3/8 & -6/8 & 3/8 & 3/8 \\ -1/8 & 2/8 & -1/8 & -1/8 \\ 2/8 & -4/8 & 2/8 & 2/8 \\ 1/8 & -2/8 & 1/8 & 1/8 \end{bmatrix}$$

Assume

$$Z = (z1,\ z2,\ z3,\ z4)^T$$

then

$$X = \begin{bmatrix} 3 \\ -1 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 5/8 & 6/8 & -3/8 & -3/8 \\ 1/8 & 6/8 & 1/8 & 1/8 \\ -2/8 & 4/8 & 6/8 & -2/8 \\ -1/8 & 2/8 & -1/8 & 7/8 \end{bmatrix} \begin{bmatrix} z1 \\ z2 \\ z3 \\ z4 \end{bmatrix} \qquad (3.25)$$

Writing equation (3.22) as a quadratic form gives

$$F = X^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X = (4) \qquad (3.26)$$

Substituting for X^T and X from equation (3.25) in equation (3.26) gives the general solution in terms of z1, z2, z3, z4. Since the vector Z is arbitrary, then for a special choice of $Z = (0, 0, 0, 0)^T$, the common zero is

$(x,y,u,w) = (3,-1,2,1)$.

3.3.2 An Application in Iterative Methods

One of the best-known methods for solving a single nonlinear equation in a single variable, say

$$f(x) = 0 \qquad (3.27)$$

is Newton's (also Newton-Raphson) method

$$x_{k+1} = x_k - [\, f(x_k)/f'(x_k) \,] \quad ,k = 0,1,... \qquad (3.28)$$

Under suitable conditions on the function f and the initial approximation x_o, the sequence (3.28) converges to a solution of (3.27); see,e.g., Ortega and Rheinboldt (11) , for iterative methods in nonlinear analysis, and in particular, for the many variations and extensions of Newton's method.

Newton's method for solving a system of m equations in n-variables

$$\begin{aligned} f_1(x1,x2,.......,xn) &= 0 \\ f_2(x1,x2,.......,xn) &= 0 \\ &\cdots\cdots \\ f_m(x1,x2,.......,xn) &= 0 \end{aligned} \qquad (3.29)$$

is similarly given, for the case m = n, by

$$x_{k+1} = x_k - [f(x_k)/f'(x_k)] \quad ,k = 0,1,... \tag{3.30}$$

where $f'(x_k)$ is the derivative of f at x_k, represented by the matrix of partial derivatives (the Jacobian matrix)

$$f'(x_k) = \partial f_i(x_k)/\partial x_j \tag{3.31}$$

If the nonsingularity of $f'(x_k)$ cannot be assumed for every x_k, and in particular, if the number of equations (3.29) is different from the number of unknowns, then it is natural to inquire whether a generalized inverse of $f'(x_k)$ can be used in (3.30), still resulting in a sequence converging to a solution of (3.29).

Ben-Israel (12) illustrates the use of generalized inverses in a modified Newton method for solving the nonlinear equations. Other applications of generalized inverses in the iterative methods of nonlinear analysis are in (12).

IV. Applications of Matrix Equations to Constrained Optimization

The nonlinear programming problem can be defined as follows:

$$\min f(X)$$

subject to

$$g_i(X) = 0 \quad , i = 1,2,\ldots,m < n,$$

$$h_j(X) \geq 0 \quad , j = 1,2,\ldots,r$$

where $X = (x1,x2,\ldots,xn)^T$ and all functions f, g_i and h_j are differentiable. Note that m, the number of equality constraints (m) , must be strictly less than n, the number of variables. If m = n the problem is termed overconstrained, since there are no degrees of freedom left for optimizing. Note that the inequality sign in $h_j(X) \geq 0$ can be reversed by multiplying through by -1 without changing the mathematical statement of the problem.

In this chapter we will consider only the case of equality constraints. Note that the inequality constraints can be changed to equality ones by adding (subtracting) slack (surplus) variables. The solution first proposed by Lagrange in 1760 was to form a new unconstrained problem by appending the constraints to the objective function with Lagrange multipliers w_i, i = 1,2,...,m. The new objective function L(X,w) is called the Lagrangian. It is defined on E^{m+n}, which is a higher-dimensional problem than the original, since it has m + n unknowns. The price paid for disposing of the constraints is this higher problem dimensionality. Since the problem defined by

$$L(X,w) = f(X) + \sum_{i=1}^{m} w_i \, g_i(X)$$

is now unconstrained, we can apply the necessary conditions for stationarity given by:

$$\partial L/dx_j = \partial f/dx_j + \sum_{i=1}^{m} w_i \, \partial g_i/\partial x_j = 0 \quad ,j = 1,2,\ldots,n,$$

$$dL/dw_i = g_i(X) = 0 \quad ,i = 1,2,\ldots,m.$$

which yield a set of m + n equations in m + n unknowns (X,w) to be solved for the optimal values (X^*,w^*).

4.1 Riccati-type Matrix Equations

This section deals with the solution of the set of non-linear algebraic equations related to the Riccati-type matrix equations. The use of matrix equations will avoid the iterative procedures. To obtain these solutions we use results of Jones (13) to obtain solutions of matrix equations of the following quadratic form:

$$XDX + AX + XB + C = 0 \qquad (4.1)$$

in which X,A,B,C and D are nxn matrices having elements belonging to the field C of complex numbers, and X is unknown.

Let R and F(R) be defined by

$$R = \begin{bmatrix} -B & D \\ -C & A \end{bmatrix} \quad , F(R) = \begin{bmatrix} U & M \\ V & N \end{bmatrix} \qquad (4.2)$$

where R is 2nx2n matrix and U,V,M and N are polynomials in

A,B,C and D. Let also $\bar{U}$ be the generalized inverse of the matrix U. The following theorems are from Jones (13).

<u>Theorem 4.1</u>

Let F(q) be any polynomial of degree $n \geq 1$ in q with coefficients belonging to C such that R and F(R) are given by (4.2). Then a solution of

$$(X,I)F(R) = (0,0)$$

with U^{-1} or M^{-1} existing, or a solution of

$$F(R)\begin{bmatrix} I \\ -X \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

with M^{-1} or N^{-1} existing is also a solution of (4.1).

Theorems (4.2) and (4.3) establish other sufficient conditions for the existance of solutions of equation (4.1).

<u>Theorem 4.2</u>

Let R and F(R) be given as in (4.2) where M^{-1} exists and

$$V = V\bar{U}U \quad ,$$

$$V\bar{U} = NM^{-1}$$

then, $X = -NM^{-1}$ is a solution of equation (4.1).

<u>Theorem 4.3</u>

Let R and F(R) be given as in (4.2) where M^{-1} exists and

$$V = N\bar{N}V \qquad , \qquad U = M\bar{N}V$$

then, $X = M^{-1}U$ is a solution of equation (4.1).

Example 4.1

Solve the following system of nonlinear equations:

$$\begin{aligned} -yz - 6x - 2y - 2z &= 0 \\ -yw - 2y - 2w + 2x &= 0 \\ -wz + 2x - 2z - 2w &= 0 \\ -w^2 + 2y + 2w + 2z &= 0 \end{aligned} \qquad (4.3)$$

This system of equations can be put in the general form given by equation (4.1) as follows:

$$X\begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}X + \begin{bmatrix} -3 & -2 \\ 2 & 1 \end{bmatrix}X + X\begin{bmatrix} -3 & 2 \\ -2 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

This gives

$$R = \begin{bmatrix} 3 & -2 & 0 & 0 \\ 2 & -1 & 0 & -1 \\ 0 & 0 & -3 & -2 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

Let

$$X = \begin{bmatrix} x & y \\ z & w \end{bmatrix}$$

the characteristic equation is

$$\det(R - qI) = (q - 1)^2(q + 1)^2 = 0$$

Table II shows that there are 3 combinations of the characteristic roots to be checked.

TABLE II

Combinations of the chacteristic roots

root	1	-1
1	(q-1)(q-1)	(q+1)(q-1)
-1	(q+1)(q-1)	(q+1)(q+1)

Case 1:

$$F(q) = (q-1)(q-1) = q^2-2q+1$$

$$F(R) = R^2-2R+I$$

$$= \begin{bmatrix} 0 & 0 & 0 & 2 \\ 0 & 0 & -2 & 2 \\ 0 & 0 & 12 & 8 \\ 0 & 0 & -8 & 4 \end{bmatrix} = \begin{bmatrix} U & M \\ V & N \end{bmatrix} \qquad (4.4)$$

Since M^{-1} exists, then by theorem (4.1) the solution of

$$(X\ ,\ I)F(R) = (0\ ,\ 0)$$

will be a solution of equation (4.3). Using equation (4.4) gives,

$$(X\ ,\ I)F(R) = (X\ ,\ I)\begin{bmatrix} U & M \\ V & N \end{bmatrix} = (0\ ,\ 0)$$

which gives the following pair of equations:

$$XU + V = 0\ , \qquad (4.5)$$

$$XM + N = 0 \qquad (4.6)$$

Since equation (4.4) gives

$$U = V = 0$$

then equation (4.5) can be satisfied by any solution of equation (4.6) given by

$$X = -NM^{-1} = \begin{bmatrix} -10 & 6 \\ 5 & 4 \end{bmatrix} \qquad (4.7)$$

Also, since N^{-1} exists, then by theorem (4.1), the solution of

$$F(R)\begin{bmatrix} I \\ -X \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

will be a solution of equation (4.3). But

$$F(R)\begin{bmatrix} I \\ -X \end{bmatrix} = \begin{bmatrix} U & M \\ V & N \end{bmatrix} \begin{bmatrix} I \\ -X \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

gives the pair of equations

$$U - MX = 0 , \tag{4.8}$$

$$V - NX = 0 \tag{4.9}$$

since

$$U = V = 0$$

then the solution of equation (4.8) and equation (4.9) is

$$X = 0$$

Case 2:

$$F(q) = q^2 - 1$$

$$F(R) = R^2 - I$$

$$= \begin{bmatrix} 4 & -4 & 0 & 2 \\ 4 & -4 & -2 & 0 \\ 0 & 0 & 4 & 4 \\ 0 & 0 & -4 & -4 \end{bmatrix} = \begin{bmatrix} U & M \\ V & N \end{bmatrix}$$

Following the same steps as in case 1, we obtain the solution

$$X = \begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix}$$

Case 3:

$$F(q) = q^2 + 2q + 1$$

$$F(R) = R^2 + 2R + I$$

$$= \begin{bmatrix} 12 & -8 & 0 & 2 \\ 8 & -4 & -2 & -2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} U & M \\ V & N \end{bmatrix}$$

By following same steps as in case 1,gives

$$X = \begin{bmatrix} -10 & 6 \\ 6 & -4 \end{bmatrix} \tag{4.10}$$

Thus the solutions of the system of equations given by equation (4.3) are the values of X obtained from the three cases.

4.2 The General Matrix Equations

In the case of the Riccati-type matrix equations, it is easy to find all the solutions by direct application of theorem (4.1). For the case of the general matrix equations of the form:

$$AX + XB + C + XDX + XEXFX = 0 \tag{4.11}$$

in which X, A, B, C, D, E, and F are nxn matrices, it is difficult till now to find all the solutions. This section deals with the problem of finding only some of the solutions of the system of equations given by (4.11).

Theorem 4.2

Let F(q) be any polynomial of degree $n \geq 1$ in q with coefficients belonging to C such that R and F(R) are given by equation (4.2). Then, X is a solution of the pair of equations:

$$\begin{gathered} DV - DNX - UEXFX = 0 \quad \text{and} \\ U = MX \end{gathered} \tag{4.12}$$

iff it is a solution of equation (4.11), provided that M^{-1} exists.

Proof:

Let X be a common solution of equation (4.12), F(R) is given by equation (4.2) and M^{-1} exists. Since F(R) is a polynomial in R and F(R) commutes with R [see (14)], then

$$R.F(R) = F(R).R$$

that gives

$$\begin{bmatrix} -B & D \\ -C & A \end{bmatrix}\begin{bmatrix} U & M \\ V & N \end{bmatrix} = \begin{bmatrix} U & M \\ V & N \end{bmatrix}\begin{bmatrix} -B & D \\ -C & A \end{bmatrix} \tag{4.13}$$

Matching the corresponding elements in the product of the matrices gives

$$\begin{aligned} -BU + DV &= -UB - MC \\ -BM + DN &= UD + MA \\ -CU + AV &= -VB - NC \\ -CM + AN &= VD + NA \end{aligned} \tag{4.14}$$

Using the identities (4.13) the following is obtained:

$$\begin{aligned} 0 &= DV - DNX - UEXFX \\ &= BU - MC - UB - DNX - UEXFX \\ &= BMX - MC - MXB - DNX - UEXFX \\ &= BMX - MC - MXB - (UD + MA + BM)X - UEXFX \\ &= -UDX - MC - MXB - MAX - UEXFX \\ &= -MXDX - MC - MXB - MAX - MXEXFX \\ &= -M(XDX + C + XB + AX + XEXFX) \end{aligned}$$

Since M^{-1} exists, then

$$AX + XB + C + XDX + XEXFX = 0 \tag{4.15}$$

Starting with equation (4.15) and reversing the steps using

$$U = MX$$

we get the system of equations given by (4.12) which completes the proof of the theorem.

Theorem 4.3

Let $F(q)$ be any polynomial of degree $n \geq 1$ in q with coefficients belonging to C, such that R and $F(R)$ are given by equation (4.2). Then X is a solution of the pair of equations:

$$-XMC + XEXFX - NC = 0 \quad \text{and} \qquad (4.16)$$
$$XU + V = 0$$

iff it is a solution of equation (4.11), provided that U^{-1} exists.

Proof:

Let X be a solution of (4.14), $F(R)$ is given by (4.2), and U^{-1} exists. Since $F(R)$ is a polynomial in R and $F(R)$ commutes with R, then (4.12), (4.13) and (4.14) hold. Using the identities (4.14) with (4.16) gives:

$$\begin{aligned}
0 &= -(XM + N)C + XEXFV \\
&= -XUB - XMC + XUB - NC + XEXFV \\
&= -X(UB + MC) + XUB - NC + XEXFV \\
&= -X(BU - DV) + XUB - NC + XEXFV \\
&= -XBU + XD(-XU) + XUB - NC + XEXFV \\
&= -XDXU - AXU - XBU + AXU + XUB - NC + XEXFV \\
&= -XDXU - AXU - XBU - AV - VB - NC + XEXFV \\
&= -XDXU - AXU - XBU - CU + XEXFU \\
&= -(XDX + AX + XB + C + XEXFX)U
\end{aligned}$$

Since U^{-1} exists, then

$$AX + XB + C + XDX + XEXFX = 0 \qquad (4.17)$$

Starting with (4.17), conversing the steps and using

$$XU + V = 0$$

we get the system of equations given by (4.16) which completes the proof of the theorem.

Example 4.2

Find the common solutions of the system of equations:

$$AX + XB + C + XDX + XEXFX = 0$$

where

$$A = B = \begin{bmatrix} 2 & 0 \\ 0 & -1 \end{bmatrix},$$

$$E = -D = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} \quad \text{and} \quad F = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}$$

Solution:

$$R = \begin{bmatrix} -B & D \\ -C & A \end{bmatrix} = \begin{bmatrix} -2 & 0 & -1 & -1 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 2 & 0 \\ 0 & -2 & 0 & -1 \end{bmatrix} \qquad (4.18)$$

the characteristic function is

$$\begin{aligned} |R - qI| &= (q^2 - 3)(q^2 - 4) \\ &= (q - 3^{1/2})(q + 3^{1/2})(q - 2)(q +- 2) \end{aligned}$$

Table III gives the all possible combinations of roots for the polynomials in q of degree 2.

TABLE III

Combinations of characteristic roots

	$q-\sqrt{3}$	$q+\sqrt{3}$	$q-2$	$q+2$
$q-\sqrt{3}$	$q^2-2\sqrt{3}q+3$	q^2-3	$q^2-(2+\sqrt{3})q+2\ 3$	$q^2+(2-\sqrt{3})q-2\sqrt{3}$
$q+\sqrt{3}$		$q^2+2\sqrt{3}q+3$	$q^2-(2-\sqrt{3})q-2\ 3$	$q^2+(2+\sqrt{3})q+2\sqrt{3}$
$q-2$			q^2-4q+4	q^2-4
$q+2$				q^2+4q+4

Let us try one of the combinations. Choose

$$F(q) = q^2 - 4$$

then,

$$F(R) = R^2 - 4I$$

$$= \begin{bmatrix} 0 & 2 & 0 & 3 \\ 0 & -1 & -3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 \end{bmatrix} = \begin{bmatrix} U & M \\ V & N \end{bmatrix}$$

Since M^{-1} exists, applying theorem (4.2) gives

$$U = MX$$

$$\begin{bmatrix} 0 & 2 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 3 \\ -3 & 0 \end{bmatrix} X$$

which gives the solution

$$X = \begin{bmatrix} 0 & 1/3 \\ 0 & 1/3 \end{bmatrix} \tag{4.19}$$

Since X given by equation (4.19) satisfies

$$DV - DNX - UEXFX = 0$$

then this solution is a solution to the given problem.

Trying other combinations can lead to some other solutions of the problem. The difficulity here is that this technique does not guarantee to find all solutions for the general matrix equations.

V. A Random Search Algorithm

The essence of this chapter is the development of a multi-dimensional random search algorithm. The technique will solve both integer and continuous nonlinear optimization problems. There are many optimization procedures which enable one to find the minimum of a unimodal function in n-space. If the function is differentiable, global minimum may be obtained through the use of derivatives. However, the problem of global optimization of multimodal function has received comparatively little attention, more so when the function in question is non-differentiable.

As a general principle, the accuracy with which a procedure locates optima improves with the number of functional evaluations. In principle, however, one seeks a balance between a degree of certainty and the cost of implementation. A procedure which locates optima with great precision and certainty would be practically worthless if it requires economically unfeasible number of calculations.

There are several search methods presently utilized to seek global optima. Borah (15) compared the method of random search with the method of systematic search and concluded that the random search method is better. His conclusion was based on the fact that the systematic search suffers from the stand point of being computationally uneconomical in that the number of evaluations increases geometrically with an increase in the number of variables. Also the method of systematic search does not, in general, obtain all the local minima. This, in turn, may lead to some doubt as to where the actual global minimum occurs. Among the random search methods, are those suggested by Brooks (16), Becker (17) and Price's CRS method (18). The simple random method accepts the optimum function value as global optimum after making a specified number of trials randomly selected from the domain. The stratified random search method divides the domain into a number of sub-domains of equal size and selects, at random, a trial point from each subdomain and each time keeps the optimal function value. The procedure is repeated many times. Some improvement on the simple random search is provided by Becker (17). His procedure begins with a simple random search over the domain. Instead of retaining the single point with the optimal function value, he retains a predetermined number of points with optimal function values in each trial. If the number of trials is sufficiently high, the retained points tend to cluster around some optima. Then a mode seeking algorithm is used to group the points into discrete clusters and to define the boundaries of the subregions each embracing a cluster. The clusters are graded, by searching in each for the retained points with the lowest function value and then rated according to the relative values of the cluster minima. The entire procedure is then repeated using as the initial search region that subdomain, defined by the mode seeking algorithm around the best cluster. The user may choose to examine also the second best cluster, or indeed all clusters, according to the extent of his doubt as to whether or not the global minimum will be found in the subdomain defined by the best cluster. Price (18) suggested the controlled random search (CRS) that combines the random search and mode-seeking algorithm into a single continuous process.

This chapter studies the problem of obtaining global optima of the general functions (differentiable or not) of several variables. The procedure begins with evaluating the given function at pre-determined number of points selected randomly over the closed bounded domain. Suppose M points are selected randomly over the domain and the function is evaluated at each of the M points. The minimal functional value and the point at which the minimum occurs (if the problem is one of minimization) are saved. This step is carried out N times. The resulting N points will cluster around the minima. An illustration of this aspect is shown in Figure (4). Suppose there are many cluster points, then there is a possibility that around each cluster point, a local minimum may exist. We develop a single program to find all the cluster groups as well as cluster points. Using a local optimization routine gives the exact minimum of each cluster. Thus the global minimum is obtained by simple comparison of the obtained exact minima.

If some of minima lie near to each other, this procedure cannot separate the clusters, because the radius of the hypersphere which embrace these cluster points should be very small and therefore many points still remain outside of any of the

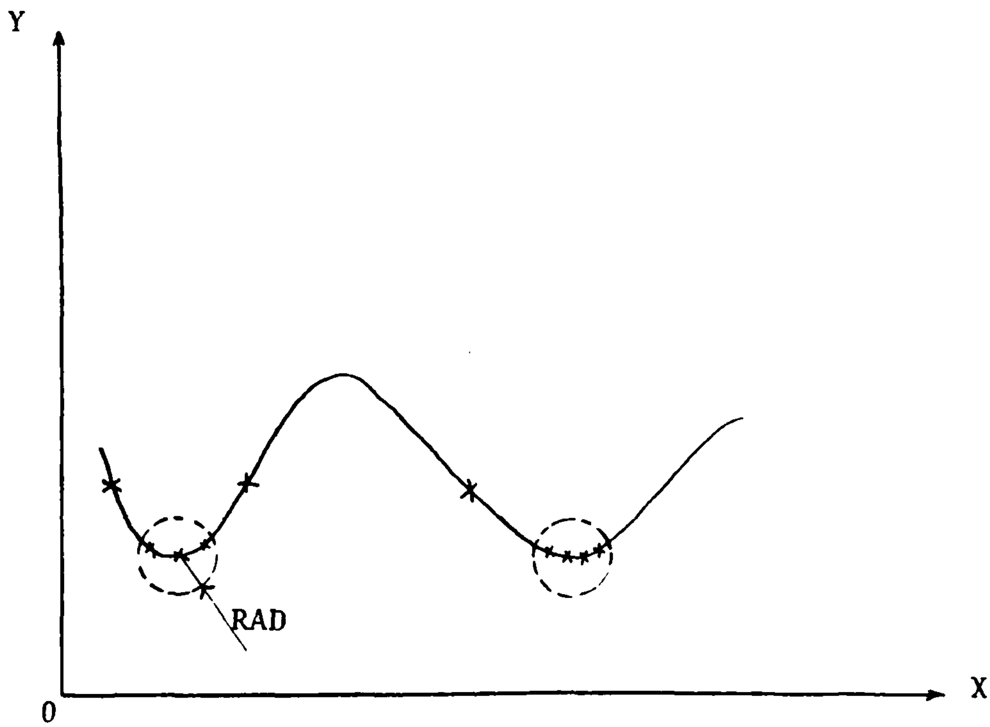


Figure 4. Formation of Clusters

hyperspheres. These points which are outside, give false cluster groups and thereby increase the function evaluations later tremendously.

After separating the cluster, the next task is to find the actual minimum in each cluster group. Any local optimizing method may be used. However, Nedler and Mead Simplex Search Method (19) is the most efficient one for the nondifferentiable functions. This simplex algorithm requires n+1 points for n-dimensional space.

5.1 The Nonlinear Simplex Method

The nonlinear simplex method,not to be confused with the simplex algorithm of linear programming, is a direct search, descent method. It is based on a geometric construct referred to,in the case of an n-dimensional space, as an n-dimensional simplex. The simplex method for minimization may be summarized as follows:

1. Construct a regular simplex in the parameter space of the variables of optimization and evaluate the objective function at each vertex.
2. Find the centroid of the simplex without the worst vertex (worst vertex being the vertex with the highest objective function value).
3. Define a new simplex by eliminating the worst vertex and adding a new vertex obtained by reflecting the worst vertex through the centroid. Evaluate the objective function at the new vertex and return to step 2.

The elegant simplicity of this algorithm has led to a large number of variations on the basic method. In 1964 Nelder and Mead extended the simplex technique by allowing irregular simplices (line segments not necessarily all of the same length). This provided both a degree of scale invaring and allowed for a form of acceleration in the search. Their modifications to the simplex rules include:

(a) If the reflected vertex is the best vertex (best vertex being the vertex with the lowest objective function value) then try an expansion step to another vertex that is further along in the same direction that yielded the reflected vertex.

(b) If the reflected vertex has the worst objective function value, then try another vertex that is retracted towards the centroid.

(c) If the retracted vertex has the worst objective function value, then contract the entire simplex by moving towards the best vertex.

(d) Stop the procedure when the simplex shrinks to a sufficiently small size.

5.2 The Nonlinear Integer Search

The development of methods for integer variable programming was initiated in the field of linear programming, and most work has continued to the general application within this area. In the field of nonlinear integer programming substantially less seems to have been done. The majority of the work that have been proposed in the literature to date are centered around one of the following four basic concepts:

(a) Rounding-off the continuous optima: The most common approach to nonlinear discrete-value programming problems is to treat the variables as continuous; then, once the continuous optimum has been determined, an additional search is executed to find a feasible set of discrete-valued variables. The most basic procedure is to select the feasible set of variables nearest to the optimal point. It is well known that this procedure can lead to undesirable answers and the point selected may not represent the discrete optimum.

(b) Adaptation of nonlinear optimization techniques: In many cases, nonlinear discrete valued programming problems are considered as a class of nonlinear programming problems in which the discreteness of variables is one of the restrictions. Many nonlinear programming techniques of this type have been devised, mostly for solving engineering design problems.

(c) Linear approximation and binary representation: Another class of approaches considers the discrete nonlinear problem as a primarily integer programming problem with nonlinear characteristics that may be linearized. Some of the approaches use techniques involving piecewise linear approximation. Others involve the transformation of a nonlinear function into a polynomial function of binary variables and then the transformation of the polynomial function into a linear function of binary variables. Several integer linear programming techniques have also been devised for solving nonlinear integer problems directly, usually after conversion to binary variable problems.

Some of these techniques, assume separable functions with certain monotonicity properties.

(d) Direct search methods: Nonlinear discrete search methods represent another class of approaches which are different from all the methods stated before. Since nonlinear discrete search occurs only over the set of points, it requires the function values at these points only. A reliable discrete search technique, however, is not easily devised due to resolution ridge difficulties and no procedure demonstrated to be reasonably reliable has appeared in the literature.

5.3 Problem Formulation

The new search technique, to be presented will consider the functions defined on a closed and bounded domain. The constraints are of the type which serve to bound the individual variables. That is:

$$a_i \leq x_i \leq b_i$$

where, x_i is the ith decision variable.

A unique feature of this algorithm is its ability to handle problems of either continuous variables or integer variables. The algorithm has, therefore, been named COIRS, an acronym for Continuous Or Integer Random Search. Most of the existing techniques for solving nonlinear integer variables programming problems were designed to solve only some particular problems with specific structures. The objective function must be expressed analytically and, in addition, many techniques contain several special requirements on the nature of the function such as continuity, differentiability or concavity. Moreover, the procedures are often exceedingly complicated which make them difficult to understand and to program for the computer.

The formulation of the optimization problem to be solved by the presented algorithm requires the objective function to be expressed analytically. The problem of V variables, P equality constraints and Q inequality constraints could be stated in one of the following forms:

(a) For continuous variables:

$$\begin{aligned} &\text{Minimize:} \quad F(X) \\ &\text{subject to:} \quad a_i \leq x_i \leq b_i \qquad ,i = 1,2,\ldots,V \\ &\qquad\qquad\quad G_j(X) = 0 \qquad\quad ,j = 1,2,\ldots,P \\ &\qquad\qquad\quad H_k(X) \geq 0 \qquad\quad ,k = 1,2,\ldots,Q \end{aligned}$$

where, $X = (x_1, x_2, \ldots, x_V)$

(b) For integer variables:

The same objective function and constraints as in (a) are used in addition to the constraint

$$x_i \text{ integers}, \quad i = 1,2,\ldots,V$$

If bounds on some variables are not given in the problem, "artificial" bounds are supplied in a way that reflects the user's guess of these bounds.

5.4 The COIRS Algorithm

Step 1: Initialization.

Step 2: Input data.

Step 3: Generate predetermined number of random points M (inside the domain) for one random search.

Step 4: Evaluate the function at all the generated points.

Step 5: Find the minimum of the evaluated functions.

Step 6: If the number of the random searches done is less than the predetermined number of random searches N, go to Step 3.

Step 7: Find the lowest minimum of the N minima determined in step 5.

Step 8: Form a cluster from some of the minima determined in Step 5 inside a hypersphere with center at the lowest minimum not yet included in any cluster.

Step 9: If the number of points in the clusters formed is less than M, go to Step 8.

Step 10: Use a local optimization technique to find the exact minimum near the center of each cluster.

Step 11: Find the global minimum by comparing the minima in Step 10.

Step 12: Stop.

Remarks:

(1) The generation of points in Step 5 is done by a random generator for both continuous and integer numbers inside the domain of the problem.

(2) The local optimization technique used in Step 10 is the Nelder and Meads Simplex technique that gives a continuous minimum. For the case of integer problems the technique is modified to find the lowest value of the function at integer hypercube around the continuous vertices. The number of vertices of the hypercube are 2^V, where V is the number of variables. For example, if the continuous vertex exists at

$$X = (1.2,\ 3.7,\ 4.9)$$

the vertices of the hypercube will be

$$X_1 = (1,\ 3,\ 4)$$
$$X_2 = (2,\ 3,\ 4)$$
$$X_3 = (1,\ 3,\ 5)$$
$$X_4 = (2,\ 3,\ 5)$$
$$X_5 = (1,\ 4,\ 4)$$
$$X_6 = (2,\ 4,\ 4)$$
$$X_7 = (1,\ 4,\ 5)$$
$$X_8 = (2,\ 4,\ 5)$$

The vertex with the lowest value of the function will enter the simplex in the place of the point that has the worst value of the function.

(3) The constraints are supplied by the user in a subroutine that determines if the generated points in Step 3 satisfy the constraints or not.

The program is written in FORTRAN V and is attached in Appendix .

5.5 Testing the Algorithm

The procedure of testing a new search technique is to demonstrate that the new technique will solve the test problems solved by the established techniques. To demonstrate the ability of the new technique one may then present a test problem that can not be solved by the established techniques but can be solved by the new COIRS technique, or one may instead solve the established test problems using fewer objective function evaluations.

The following three examples are from Borah (15):

Example 5.1

$$\text{Minimize } f(x,y,z) = (x - y + z)^2 + (-x + y + z)^2 + (x + y - z)^2$$

$$\text{subject to:} \quad -1 \leq x,y,z \leq 1$$

The actual solution of this problem is:

$$f = 0$$

$$\text{at} \quad x = y = z = 0$$

The solution by the COIRS algorithm after 1962 functional evaluations is:

$$f = .4466\text{x}10^{-7}$$

$$\text{at} \quad x = .000113$$

$$y = .0001585$$

$$z = -.00002563$$

Example 5.2

$$\text{Minimize } f(x,y,z) = 9 - 8x - 6y - 4z + 2x^2 + 2y^2 + 2z^2 + 2xy + 2xz$$

subject to: $0 \leq x,y,z \leq 1.5$

The actual solution of this problem is:

$$f = 0$$

at $x = y = z = 0$

A comparison of the solution given by COIRS algorithm and Borah's solution (15) is shown in Table IV.

TABLE IV

Comparison of COIRS with Borah's Results

	COIRS	Borah
Number of functional evaluations	1826	14144
f	$.6537 \times 10^{-7}$	$.953674 \times 10^{-4}$
x	1	1.013671875
y	1	.9921875
z	.999	.986328123

Example 5.3

$$\text{Minimize } f(x,y,z) = 9 - 8x - 6y - 4z + 2x^2 + 2y^2 + 2z^2 + 2xy + 2xz$$

subject to: $0 \leq x,y,z \leq 1.5,$

x,y,z integers

The actual integer solution of this problem is:

$$f = 0$$

at $\quad x = y = z = 1$

Using the COIRS algorithm for the case of integer variables, the solution after 800 functional evaluations is:

$$f = 0$$

at $\quad x = y = z = 1$

<u>Example</u> <u>5.4</u>

Minimize $f(x,y,z) = 100[z-(x+y)^2/4]^2 + (1-x)^2 + (1-y)^2$

subject to: $0 \leq x,y,z \leq 1.5$,

x,y,z integers

The actual solution for this problem is:

$$f = 0$$

at $\quad x = y = z = 1$

Using the COIRS algorithm for the case of integer variables, the solution after 540 functional evaluations is:

$$f = 0$$

at $\quad x = y = z = 1$

To complete the testing of the algorithm, many other problems have to be solved and compared with the solutions by other algotithms. This may indicate additional features of the COIRS algorithm.

VI. Conclusions and Directions for Further Work

There are several computer applications of iterative techniques which will iterate towards zeros of a function. The initial starting value for these schemes must be chosen carefully. If the initial point is not close to an actual zero, the iterative techniques often diverge.

The ease of performing the calculations given by the functionals on the trial functions makes these techniques for determining whether a function vanishes or is bounded in a certain region attractive. Other choices of trial functions could be made, perhaps giving better results in certain situations than other trial functions.

The theorems and examples presented in this work in chapters 2 and 3 can be used to locate intervals on which a function has a zero. If the given function can be paired with one of the special differential equations given, and if a trial function u is found which makes the corresponding functional J negative on a certain interval, then a zero of the function f does exist on that interval. These techniques have a possible use in determining an initial guess for a starting point in iterative search techniques.

Examples of common functions from different engineering sciences were chosen to demonstrate the breadth of applications possible. Perhaps this method can be applied to indicate the existance of zeros of other functions before valuable computer time is wasted to find these zeros. In fact, the techniques obtained can themselves be implemented using numerical methods on a computer of reasonable size.

There is no doubt about the results of techniques presented in chapters 2 and 3 for the unimodal functions. For the case of multimodal functions further studies and analysis is required. Application of given theorems for the case of multimodal functions will result in dividing the region of the function into subregions with at least one zero on each subregion. A comparison of this technique with the known search techniques is an area for further study.

Analysis presented in chapters 2 and 3 is limited to some functions that satisfy special classes of homogeneous differential equations. The case of nonhomogeneous differential equations and other classes of differential equations needs further analysis and effort. Other remaining aspect in this area is to compare our techniques with other approaches that can be used to solve the problem of finding good initial point for iterative methods.

For the case of constrained optimization, the technique of generalized inverse was given with illustrative examples in chapter 3. The generalized inverse deals with the special case when the problem is ill conditioned. This technique is applied to one of the best-known methods for solving a system of nonlinear equations in n-variables. The problem appears when the matrix of partial derivatives (the Jacobian matrix) is singular.

Applications of matrix equations to constrained optimization problems is presented in chapter 4. For the special case of the Riccati-type matrix equations there is no problem for finding all the solutions of the problem by applying the technique of matrix equations. The difficulity appears in the case of the general matrix equations where only some of the solutions can be obtained by this technique. The problem of finding all the solutions in the general case is a wide area for further study and analysis. Some theorems have to be developed to guarantee finding all the solutions.

The general optimization problem for continuous or integer variables is covered in chapter 5. The COIRS algorithm is tested by some examples. Still, it needs further testing with different problems. The case of mixed integer problems using the COIRS algorithm is a recommended area of further study and research. An algorithm that solves the general mixed integer problem without any restrictions on the objective function will be a superior one in optimization.

Appendix A

Computer Print Out to Find H

```
      PROGRAM RUMBRG
      ****************************************************************
      THIS PROGRAM FINDS THE VALUE OF H AT WHICH THE FUNCTIONAL
      J IS NEGATIVE.  THE CALCULATION OF J IS MADE BY ROMBERG
      METHOD.
            A = LOWER LIMIT OF X
            B = UPPER LIMIT OF X
            H = PARAMETER
            D = INCREMENT OF H
       F(H,X) = THE INTEGRAND OF J
      ****************************************************************
      DIMENSION T(10,10)
      KMAX=6
      A=0.
      B=100.
      H=.1
      D=.001
    1 M=2
      C=(B-A)/FLOAT(M)
      SUM=(F(H,A)+F(H,B))/2.
      MM1=M-1
      IF(MM1)40,10,8
    8 DO 9 I=1,MM1
    9 SUM=SUM+F(H,A+FLOAT(I)*C)
   10 T(1,1)=SUM*C
      DO 20 K=2,KMAX
      C=C/2.
      M=M*2
      SUM=0.
      DO 11 I=1,M,2
   11 SUM=SUM+F(H,A+FLOAT(I)*C)
              T(K,1)=T(K-1,1)/2.+SUM*C
      FOURJ=1.
      DO 12 J=2,K
      FOURJ=FOURJ*4.
      SAVE DIFFERENCES FOR LATER CALC.OF RATIOS
      T(K-1,J-1)=T(K,J-1)-T(K-1,J-1)
      T(K,J)=T(K,J-1)+T(K-1,J-1)/(FOURJ-1.)
      IF(T(K,J).LT.0.0000001) GO TO 2
   12 CONTINUE
   20 CONTINUE
      IF(H.GE.B) GO TO 40
      IF(H.LT.B) H=H+D
      GO TO 1
    2 PRINT*,'H='
      PRINT*,H
```

```
   40 STOP
      END
      FUNCTION F(H,X)
      F=(H-2*X)**2-(H*X-X*X)
      RETURN
      END
```

Appendix B

Computer Print Out to Find H_1 and H_2

```
      PROGRAM MULINT
      REAL X,Y,A,B,U1,U2,H1,H2,C,Z
      COMMON H1,H2
      U1=10.
      U2=10.
      H1=.1
      H2=.1
      D1=.001
      D2=.001
      A=0.
    1 C=0.
      N=5
      M=5
      NN=2*N+1
      MM=2*M-1
      H=(H1-A)/(2*N)
      AN=0.
      AE=0.
      AO=0.
      DO 10 I=1,NN
         X=A+(I-1)*H
         HX=(H2-C)/(2*M)
         BN=F(X,C)+F(X,H2)
         BE=0.
         BO=0.
         DO 20 J=1,MM
            Y=C+J*HX
            Z=F(X,Y)
            IF(J.EQ.2*(J/2)) THEN
               BE=BE+Z
            ELSE
               BO=BO+Z
            END IF
   20    CONTINUE
         A1=(BN+2*BE+4*BO)*HX/3
         IF(I.EQ.1.OR.I.EQ.NN)THEN
            AN=AN+A1
         ELSE
            IF(I.EQ.2*(I/2)) THEN
               AO=AO+A1
            ELSE
               AE=AE+A1
            END IF
         END IF
   10 CONTINUE
      A.=(AN+2*AE+4*AO)*H/3
```

```
      IF(A2.GE.0.) THEN
      H1=H1+D1
      H2=H2+D2
      ELSE
      GO TO 25
      END IF
      IF(H1.GE.U1.OR.H2.GE.U2)GO TO 30
      GO TO 1
   25 PRINT*,H1,H2
   30 STOP
      END
      FUNCTION F(X,Y)
      COMMON H1,H2
      F=((22/7)*(1/H1)*COS((22/7)*X/H1)*SIN((22/7)*Y/H2))**2
     1+((22/7)*(1/H2)*SIN((22/7)*X/H1)*COS((22/7)*Y/H2))**2
     1-(SIN((22/7)*X/H1)*SIN((22/7)*Y/H2))**2
      RETURN
      END
```

Appendix C

Computer Print Out for COIRS Algorithm

```
      PROGRAM COIRS
C     ****************************************************************
C     THIS PROGRAM FINDS THE MINIMUM OF A FUNCTION USING A RANDOM
C     SEARCH TECHNIQUE.  THE FUNCTION CAN BE DIFFERENTIABLE OR
C     NOT.
C        THIS ALGORITHM NEEDS THE UPPER AND LOWER LIMITS OF THE
C     VARIABLES TO BE KNOWN.
C     ****************************************************************
      DIMENSION A(3,20,20),FMAX(20),CMAX(20,3),F(20,20),
     1   CF(20,3),MAX(20,20),X(20),Y(20),YMAX(20,20),
     2   CYMAX(20,10,3),FMAX1(20),A(3,10),AX1(20,3),CFAX(20,3),
     3   CL1(20,20),FCL(20),CFCL(20,3),CMAX2(20,3),FMAX2(20),
     4 CL(20,10,3),D(6),E(3),A1(3,20,20)
      INTEGER IE(3),IA(3,20,20),ID(6),ICMAX(20,3
C     ****************************************************************
C     THE DATA ARE THE UPPER AND LOWER BOUNDS OF THE VARIABLES
C     ****************************************************************
      DATA D/0.0,0.0,0.0,1.5,1.5,1.5/
      OPEN (3,FILE='IFLAG1')
      REWIND 3
C     ************************************************************
C     TO FIND THE INTEGER LIMITS OF THE VARIABLES
C     I2=THE NUMBER OF VARIABLES
C     I1=I2X2
C     ************************************************************
      I2=3
      DO 12 K=1,6
      IF(K.GE.(I2+3)) GO TO 12
      IF(AMOD(D(K),1.0).GT.0.0) ID(K)=INT(D(K))+1
   12 ID(K)=INT(D(K))
      I1=6
      FAD=.1
      KOUNT=0
C     ************************************************
C     IFLAG=1 FOR THE INTEGER VARIABLES
C          =2 FOR THE CONTINUOUS VARIABLES
C         M=THE NUMBER OF RANDOM SEARCHES
C         N=THE NUMBER OF POINTS IN EVERY SEARCH
C     ************************************************
      IFLAG=1
      M=10
      N=10
C     ***********************
C     GENERATE COORDINATES
C     ***********************
      DO 40 J=1,M,1
```

```
      DO 30 I=1,N,1
      DO 13 K=1,I2
      CALL UNFORM(L,XR,K)
      IF(IFLAG.EQ.1) GO TO 700
      GO TO 9
  700 IA(K,I,J)=INT(XR)
      IF(IA(K,I,J).LT.IB(K)) IA(K,I,J)=IA(K,I,J)+1
      AI(K,I,J)=IA(K,I,J)*1.0
      GO TO 13
    9 A(K,I,J)=XR
   13 CONTINUE
C     ****************************************
C     CALCULATE THE FUNCTIONAL VALUES
C     KOUNT=COUNTER FOR FUNCTIONAL EVALUATIONS
C     ****************************************
    4 N1=J
      K1=I
      IF(IFLAG.EQ.1) F(I,J)= CAL(AI,K1,N1)
      IF(IFLAG.EQ.2) F(I,J)=CAL(A,K1,N1)
      KOUNT=KOUNT+1
      IF(IFLAG.EQ.1) GO TO 6
      WRITE(3,26)(A(K,I,J),K=1,I2),F(I,J)
   26 FORMAT(4X,3(E10.4,4X),4X,E10.4/)
      GO TO 30
    6 WRITE(3,27)(IA(K,I,J),K=1,I2),F(I,J)
   27 FORMAT(4X,3(I12,3X),3X,E10.4/)
      DO 28 K=1,I2
   28 A(K,I,J)= IA(K,I,J)*1.0
   30 CONTINUE
C     ******************************************************************
C     CALL SUBROUTINE TO FIND MIN OF THE 10 POINTS JUST EVALUATED
C     ******************************************************************
      K=J
      I3=I2
      CALL FMIN1(A,F,FMAX,LMAX,K,N,I3)
C     ****************************************
C     CALCULATION TO START NEXT RANDOM SEARCH
C     ****************************************
      AJ=J
      AJ=AJ/10.
      DO 35 K=1,I2,1
      IF(IFLAG.EQ.1)IA(K,1,1)=IA(K,1,J)+AJ*(-1)**J
      IF(IFLAG.EQ.2) A(K,1,1)=A(K,1,J)+AJ*(-1)**J
   35 CONTINUE
   40 CONTINUE
C
C     OUTPUT MINIMUM VALUE AND COORDINATES
C
      WRITE(3,45)
```

```
   45 FORMAT(' ',5X,'VALUE OF THE FUNCTION',5X,'FIRST COORDINATE
     *',5X,'SECOND COORDINATE ',5X,'THIRD COORDINATE',/)
      IF(IFLAG.EQ.1) GO TO 56
      DO 50 I=1,N,1
   50 WRITE(3,55) FMAX(I),(CMAX(I,K),K=1,I2)
   55 FORMAT(5X,E11.5,5X,E11.5,5X,E11.5,5X,E11.5)
      GO TO 54
 56   DO 58 I=1,N,1
      DO 57 K=1,I2
 57   ICMAX(I,K)=INT(CMAX(I,K))
 58   WRITE(3,59)FMAX(I),(ICMAX(I,K),K=1,I2)
 59   FORMAT(5X,E11.5,5X,3(I11,5X))
C
C     CALL PLOTTING ROUTINE
C
   54 DO 200 J=1,4,1
      JJ=J
      CALL MIN2(CMAX,FMAX,FMAX2,CMAX2,N,JJ,I2)
   80 K=0
      DO 84 I=1,N,1
      IF(FMAX(I).EQ.9.1E+10)GO TO 84
      DIST=0.0
      DO 81 KI=1,I2,1
      P(KI)=CMAX2(J,KI)-CMAX(I,KI)
   81 DIST=DIST+(P(KI)**2)
      Q=SQRT(DIST)
      IF(Q.GE.RAD) GO TO 84
      K=K+1
      FCL(K)=FMAX(I)
      DO 82 KI=1,I2,1
   82 CFCL(K,KI)=CMAX(I,KI)
      FMAX(I)=9.1E+10
      DO 83 KI=1,I2,1
   83 CMAX(I,KI)=9.1E+10
   84 CONTINUE
      K=K+1
      FCL(K)=FMAX2(J)
      DO 85 KI=1,I2,1
   85 CFCL(K,KI)=CMAX2(J,KI)
      FMAX2(J)=9.1E+10
      IF(K.EQ.0.) GO TO 220
      GO TO(101,111,121,131)J
  101 K1=K
      DO 106 I=1,K1,1
      CL1(J,I)=FCL(I)
      FCL(I)=0.
      DO 105 KI=1,I2,1
      CC1(J,I,KI)=CFCL(I,KI)
  105 CFCL(I,KI)=0.
```

```
 106 CONTINUE
     GO TO 200
 111 K2=K
     DO 116 I=1,K2,1
     CL1(J,I)=FCL(I)
     FCL(I)=0.
     DO 115 KI=1,I2,1
     CL(J,I,KI)=CFCL(I,KI)
 115 CFCL(I,KI)=0.
 116 CONTINUE
     GO TO 200
 121 K3=K
     DO 126 I=1,K3,1
     CL1(J,I)=FCL(I)
     FCL(I)=0.
     DO 125 KI=1,I2,1
     CL(J,I,KI)=CFCL(I,KI)
 125 CFCL(I,KI)=0.
 126 CONTINUE
     GO TO 200
 131 K4=K
     DO 136 I=1,K4,1
     CL1(J,I)=FCL(I)
     FCL(I)=0.
     DO 135 KI=1,I2,1
     CL(J,I,KI)=CFCL(I,KI)
 135 CFCL(I,KI)=0.
 136 CONTINUE
 200 CONTINUE
C
C    WRITE THE FUNCTION-VALUE AND THE COORDINATES OF THE WHOLE
C    CLUSTER
C
 220 IF(K1.EQ.0) GO TO 225
     WRITE(3,222)K1
 222 FORMAT(10X,'CLUSTER OF ',I3)
     DO 224 I=1,K1,1
 224 WRITE(3,*) CL1(1,I),(CL(1,I,KI),KI=1,I2)
 225 IF(K2.EQ.0) GO TO 230
     WRITE(3,226)K2
 226 FORMAT(10X,'CLUSTER OF ',I3)
     DO 228 I=1,K2,1
 228 WRITE(3,*) CL1(2,I),(CL(2,I,KI),KI=1,I2)
 230 IF(K3.EQ.0) GO TO 235
     WRITE(3,232)K3
 232 FORMAT(10X,'CLUSTER OF',I3)
     DO 234 I=1,K3,1
 234 WRITE(3,*) CL1(3,I),(CL(3,I,KI),KI=1,I2)
 235 IF(K4.EQ.0) GO TO 240
```

```
      WRITE(3,236)K4
  236 FORMAT(19X,'CLUSTER OF',I3)
      DO 238 I=1,K4,1
  238 WRITE(3,*)CL1(4,I),(CL(4,I,KT),KT=1,I2)
240   KK=0
      DO 350 J=1,4,1
      GO TO (302,306,310,314)J
  302 IF(K1.EQ.0) GO TO 350
      J1=K1
      I=J
      ICOUNT=0
  303 CALL SMPLEX(CL1,CL,CYMAX,YMAX,N,I,J1,I2,ICOUNT,ID)
C     ***************************************************
C     ICOUNT=COUNTER FOR THE SIMPLEX ITERATIONS
C     ******************************************************
      WRITE(3.305)ICOUNT
  305 FORMAT(2X.'GROUP ITERATION',2X,I4)
      KK=KK+ICOUNT
      GO TO 350
  306 IF(K2.EQ.0) GO TO 350
      J1=K2
      I=J
      GO TO 303
  310 IF(K3.EQ.0) GO TO 350
      J1=K3
      I=J
      GO TO 303
  314 IF(K4.EQ.0) GO TO 350
      J1=K4
      I=J
      GO TO 303
  350 CONTINUE
      KNT=KK+KOUNT
      WRITE(3,355)KNT
  355 FORMAT(10X,'TOTAL ITERATION ',2X,I5)
      STOP
      END
      SUBROUTINE MIN2(CMAX3,FMAX3,FAX2,CFAX2,M.L,II)
C     ******************************************************
C     THIS SUBROUTINE FINS THE MINIMUM OF THE M MINIMA
C     ******************************************************
      DIMENSION FAX2(20).CMAX3(20,3),FMAX3(20),CFAX2(20,3),TEM(3)
      II=3
      DO 10 I=1.M,1
      IF(FMAX3(I).GE.FMAX3(1)) GO TO 10
      TEMP=FMAX3(1)
      FMAX3(1)=FMAX3(I)
      FMAX3(I)=TEMP
      DO 5 K=1,3.1
```

```
      TEM(K)=CMAX3(1,K)
      CMAX3(1,K)=CMAX3(I,K)
      CMAX3(I,K)=TEM(K)
    5 CONTINUE
   10 CONTINUE
C     ***************************
C     STORE THE CURRENT LOWER VALUE
C     ***************************
      FMAX2(L)=FMAX3(1)
      FMAX3(1)=9.1E+10
      DO 12 K=1,3,1
      CFAX2(L,K)=CMAX3(1,K)
   12 CMAX3(1,K)=9.1E+10
      RETURN
      END
      SUBROUTINE FMIN1(B,H,FMAX1,CMAX1,L,N1,L3)
C     ********************************************
C     THIS SUBROUTINE FINDS THE MINIMUM OF N POINTS
C     ********************************************
      DIMENSION B(3,20,20),H(20,20),FMAX1(20),CMAX1(20,3),TEM(3)
      L3=3
      DO 10 I=1,N1,1
      IF(H(I,L).GE.H(1,L)) GO TO 10
      TEMP=H(1,L)
      H(1,L)=H(I,L)
      H(I,L)=TEMP
      DO 5 KK=1,L3,1
      TEM(KK)=B(KK,1,L)
      B(KK,1,L)=B(KK,I,L)
    5 B(KK,I,L)=TEM(KK)
   10 CONTINUE
C     ********************************************
C     STORE THE CURRENT LOWER VALUE AND COORDINATES
C     ********************************************
      FMAX1(L)=H(1,L)
      DO 12 KK=1,L3,1
   12 CMAX1(L,KK)=B(KK,1,L)
      RETURN
      END
C     **************************************************
C     LOCAL OPTIMIZATION USING NELDER AND MEADS SIMPLEX METHOD
C     **************************************************
      SUBROUTINE SMPLEX(ACL1,CACL1,CZMAX,ZMAX,DD,L,M,N4,NOUT,II)
      DIMENSION ACL1(20,20),CACL1(20,20,3),CZMAX(20,20,3),
     *ZMAX(20,20),TEMP(20),DD(6),SUM(3)
     *    ,X2(4,4),SUM(3),FIT(4,4),CPOT(4,4,3),CP(4,3),OP(4,3),
     *FIMG(4,3),FOR(4),FPM(4),FCN(4),FMX(4),CW(4,3),X1(4),EX(4,3)
      REAL X(3)
      INTEGER IX(3),II(6)
```

```
      IFLAG=2
      RADD=.9
      I4=3
      ITER=0.
      N2=I4+1
      DO 3 I=1,N2,1
      BST(L,I)=+.91E+10
      DO 3 K=1,I4,1
    3 CBST(L,I,K)=(+1)**(*0.91E+1)
C     ***************************************************************
C     FIND THE LOWEST POINT AND THE COORDINATES BST(L,1)
C     ***************************************************************
      DO 10 I=1,M,1
      IF(ACL1(L,I).EQ.0.91E+10) GO TO 10
      IF(BST(L,1).LE.ACL1(L,I))GO TO 10
         TEMP=BST(L,1)
      BST(L,1)=ACL1(L,I)
      ACL1(L,I)=TEMP
      DO 8 K=1,I4,1
      TEMPO(K)=CBST(L,1,K)
      CBST(L,1,K)=CACL1(L,I,K)
    8 CACL1(L,I,K)=TEMPO(K)
   10 CONTINUE
      WRITE(3,*)BST(L,1)
C     *********************
C     FIND THE SECOND BEST
C     *************************
      DO 20 I=1,M,1
      IF(ACL1(L,I).EQ.BST(L,1)) GO TO 20
      IF(ACL1(L,I).EQ.0.91E+10) GO TO 20
      IF(BST(L,2).LE.ACL1(L,I)) GO TO 20
      TEMP=BST(L,2)
      BST(L,2)=ACL1(L,I)
      ACL1(L,I)=TEMP
      DO 16 K=1,I4,1
      TEMPO(K)=CBST(L,2,K)
      CBST(L,2,K)=CACL1(L,1,K)
   16 CACL1(L,1,K)=TEMPO(K)
   20 CONTINUE
      WRITE(3,*)BST(L,2)
C     ***************************
C     FIND THE THIRD LOWEST POINT
C     ***************************
      DO 26 I=1,M,1
      IF(ACL1(L,I).EQ.BST(L,1)) GO TO 26
      IF(ACL1(L,I).EQ.BST(L,2)) GO TO 26
      IF(ACL1(L,I).EQ.0.91E+10) GO TO 26
      IF(BST(L,3).LT.ACL1(L,I)) GO TO 26
      TEMP=BST(L,3)
```

```
      ACL1(L,I)=TEMP
      DO 25 K=1,I4,1
      TEMPO(K)=CBST(L,3,K)
      CBST(L,3,K)=CACL1(L,I,K)
   25 CACL1(L,I,K)=TEMPO(K)
   26 CONTINUE
      WRITE(3,*)BST(L,3)
C     *****************************************
C     FIND THE FOURTH LOWEST POINT AT THIS TIME
C     *****************************************
      DO 29  I=1,M,1
      IF(ACL1(L,I).EQ.BST(L,1)) GO TO 29
      IF(ACL1(L,I).EQ.BST(L,2)) GO TO 29
      IF(ACL1(L,I).EQ.BST(L,3)) GO TO 29
      IF(ACL1(L,I).GE.BST(L,4)) GO TO 29
      IF(ACL1(L,I).EQ.0.91E+10) GO TO 29
      TEMP=BST(L,4)
      BST(L,4)=ACL1(L,I)
      ACL1(L,I)=TEMP
      DO 28 K=1,I4,1
      TEMPO(K)=CBST(L,4,K)
      CBST(L,4,K)=CACL1(L,I,K)
   28 CACL1(L,I,K)=TEMPO(K)
   29 CONTINUE
      WRITE(3,*)BST(L,4)
C     ************************************************************
C     THE SIMPLX FORMED BY BST(L,1),BST(L,2),BST(L,3),BST(L,4)
C     WHICH IS THE BIGGEST ONE SHOULD BE REMOVED
C     ************************************************************
   31 DO 35 K=1,I4,1
      SUM(K)=0.
      DO 35 J=1,N2,1
   35 SUM(K)=SUM(K)+CBST(L,I,K)
      DO 36 K=1,I4,1
   36 X1(K)=SUM(K)/4.
      DIF=0.
      DO 40 I=1,N2,1
      DO 39 K2=1,I4,1
      X2(I,K2)=CBST(L,I,K2)-X1(K2)
   39 DIF=DIF+X2(I,K2)**2
   40 CONTINUE
      DIFB=SQRT(DIF)
      ITER=ITER+1
      IF(DIFB.LE.RADIU) GO TO 500
      WRITE(3,43) DIFB
   43 FORMAT(4X,'DIFFERENCE=',E12.5)
C     ******************************************************
C     REMOVE THE HIGHEST FROM THE SIMPLEX
C     THE HIGHEST POINT IS THE BST(L,4)
```

```
C     THE MEDIAN OF THE POINTS BST(L,1),BST(L,2),BST(L,3)
C     ********************************************************
      DO 48 K=1,I4,1
      SUM2(K)=0.
      N3=NT-1
      DO 46 J=1,N3,1
   46 SUM2(K)=SUM2(K)+CBST(L,J,K)
   48 CP(L,K)=SUM2(K)/3.
C     *************************************
C     FIND THE IMAGE OF BST(L,4) THOUGH CP
C     *************************************
      DO 50 K=1,I4,1
   50 PIMG(L,K)=2.*CP(L,K)-CBST(L,4,K)
C     ********************************************
C     CHECK IF EXCEEDS THE DOMAIN OF THE FUNCTION
C     ********************************************
      DO 52 K=1,I4,1
      IF(PIMG(L,K).LT.DD(K))PIMG(L,K)=DD(K)
   52 IF(PIMG(L,K).GT.DD(K+3))PIMG(L,K)=DD(K+3)
C     *************************************
C     EVALUATE THE FUNCTION AT THESE POINTS
C     *************************************
      L3=L
      M1=I4
      FPMG(L)=XFCT(PIMG,M1,L3)
      KOUT=KOUT+1
      WRITE(3,54)FPMG(L)
   54 FORMAT(4X,'FPMG=',E12.4)
      IF(FPMG(L).LT.BST(L,1)) GO TO 200
      IF(FPMG(L).LT.BST(L,2)) GO TO 200
      IF(FPMG(L).GT.BST(L,4)) GO TO 60
      DO 56 K=1,I4,1
   56 CR(L,K)=(3*CP(L,K)-CBST(L,4,K))/2.
C     ************************
C     CHECK IF EXCEEDS DOMAIN
C     ************************
      DO 58 K=1,I4,1
      IF(CR(L,K).LT.DD(K)) CR(L,K)=DD(K)
   58 IF(CR(L,K).GT.DD(K+3)) CR(L,K)=DD(K+3)
C     **********************
C     EVALUATE THE FUNCTION
C     **********************
      M1=I4
      L3=L
      FCR(L)=XFCT(CR,M1,L3)
      KOUT=KOUT+1
      BST(L,4)=FCR(L)
      DO 59 K=1,I4,1
   59 CBST(L,4,K)=CR(L,K)
```

```
C      GO TO 400
   60  DO 65 K=1,I4,1
   65  CW(L,K)=(CP(L,K)+CBST(L,4,K))/2.
C      *********************
C      SEE IF EXCEEDS DOMAIN
C      *********************
       DO 70 K=1,I4,1
       IF(CW(L,K).LT.DD(K)) CW(L,K)=DD(K)
   70  IF(CW(L,K).GT.DD(K+3)) CW(L,K)=DD(K+3)
       M1= 4
       L3=L
       FCW(L)=XFCT(CW,M1,L3)
       KOUT=KOUT+1
       BST(L,4)=FCW(L)
       DO 75 K=1,I4,1
   75  CBST(L,4,K)=CW(L,K)
       GO TO 400
  100  BST(L,4)=FPMG(L)
       DO 110 K=1,I4,1
  110  CBST(L,4,K)=PIMG(L,K)
       GO TO 400
  200  DO 220 K=1,I4,1
  220  EX(L,K)=3.0*CP(L,K)-2.0*CBST(L,4,K)
C      *********************
C      SEE IF EXCEEDS DOMAIN
C      *********************
       DO 250 K5=1,I4,1
       IF(EX(L,K5).LT.DD(K5)) EX(L,K5)=DD(K5+3)
  250  IF(EX(L,K5).GT.DD(K5+3)) EX(L,K5)=DD(K5+3)
       M1=I4
       L3=L
       FEX(L)=XFCT(EX,M1,L3)
       KOUT=KOUT+1
       IF(FEX(L).LT.BST(L,1)) GO TO 310
       WRITE(I,255) FEX(L)
  255  FORMAT(10X,'FEX=',E10.4)
       GO TO 100
  310  BST(L,4)=FEX(L)
       DO 320 K=1,I4,1
  320  CBST(L,4,K)=EX(L,K)
  400  IF(IFLAG.EQ.1) GO TO 325
       GO TO 350
  325  DO 330 I=1,3
  330  X(I)=CBST(L,4,I)
       CALL PERM (X,3),FA,ID)
       DO 340 I=1,3
  340  CBST(L,4,I)=IY(I)*1.0
       BST(L,4)=FA
  350  DO 410 I=1,N2,1
```

```
      IF(BST(L,1).LT.BST(L,I)) GO TO 410
      TEMP=BST(L,1)
      BST(L,1)=BST(L,I)
      BST(L,I)=TEMP
      DO 405 K=1,I4,1
      TEMPO(K)=CBST(L,1,K)
      CBST(L,1,K)=CBST(L,I,K)
  405 CBST(L,I,K)=TEMPO(K)
  410 CONTINUE
      DO 450 I=2,N2,1
      IF(BST(L,2).LT.BST(L,I)) GO TO 450
      TEMP=BST(L,2)
      BST(L,2)=BST(L,I)
      BST(L,I)=TEMP
      DO 420 K=1,I4,1
      TEMPO(K)=CBST(L,2,K)
      CBST(L,2,K)=CBST(L,I,K)
  420 CBST(L,I,K)=TEMPO(K)
  450 CONTINUE
      IF(BST(L,3).LT.BST(L,4)) GO TO 460
      TEMP=BST(L,3)
      BST(L,3)=BST(L,4)
      BST(L,4)=TEMP
      DO 455 K=1,I4,1
      TEMPO(K)=CBST(L,3,K)
      CBST(L,3,K)=CBST(L,4,K)
  455 CBST(L,4,K)=TEMPO(K)
  460 IF(ITER.LT.10.) GO TO 31
      ITER=0.
      BST(L,4)=(BST(L,1)+BST(L,2)+BST(L,3)+BST(L,4))/4.
      GO TO 400
  500 ZMAX(L,1)=BST(L,1)
      DO 505 K=1,I4,1
  505 CZMAX(L,1,K)=CBST(L,1,K)
      WRITE(3,510) ZMAX(L,1),(CZMAX(L,1,K),K=1,I4)
  510 FORMAT(10X,E10.4,3(E10.4,4X),/)
      RETURN
      END
      FUNCTION XFCT(C,IJ,IF)
      DIMENSION C(4,3)
      II=3
      XFCT=5.0-4.0*C(IF,1)-4.0*C(IF,2)-4.0*C(IF,3)+2.0*C(IF,1)
     1**2+2.0*C(IF,2)**2+C(IF,3)**2+2*C(IF,1)*C(IF,2)*2.0-
     12.0*C(IF,1)*C(IF,3)
      RETURN
      END
      FUNCTION CAL(C1,L3,N3)
      DIMENSION C1(3,20,20)
      CAL=9.0-8.0*C1(1,L3,N3)-6.0*C1(2,L3,N3)-4.0*C1(3,L3,N3)+
```

```
      12.0*C1(1,L3,N3)**2+2.0*C1(2,L3,N3)**2+C1(3,L3,N3)**2
      1+2.0*C1(1,L3,N3)*C1(2,L3,N3)+2.0*C1(1,L3,N3)*C1(3,L3,N3)
       RETURN
       END
       SUBROUTINE PERMUT(XP,NEWC,FNEW,ID)
C      ************************************************
C      THIS SUBROUTINE GENERATES INTEGER POINTS AROUND THE
C      NEW CONTINUOUS POINT OF THE SIMPLEX
C      ************************************************
       DIMENSION XP(3),VALF(8),EX(3)
       INTEGER MARK(3),LX(3),KX(3),ICOR(8,3),NEWC(3),IFP(8),ID(6)
       DO 1 I=1,3
       MARK(I)=2
       EX(I)= AMOD(XP(I),1.0)
       IF(EX(I).EQ.0.0) MARK(I)=1
       LX(I)=INT(XP(I))
     1 KX(I)=LX(I)+1
       DO 2 I=1,3
       IF(MARK(I).EQ.2) GO TO 2
       KX(I)=LX(I)
     2 CONTINUE
       DO 3 J=1,4
       ICOR(J,1)=LX(1)
     3 ICOR(J+4,1)=KX(1)
       DO 4 J=1,2
       ICOR(J,2)=LX(2)
       ICOR(J+2,2)=KX(2)
       ICOR(J+4,2)=LX(2)
     4 ICOR(J+6,2)=KX(2)
       DO 5 J=1,7,2
       ICOR(J,3)=LX(3)
     5 ICOR(J+1,3)=KX(3)
       DO 6 L=1,8
       DO 8 K=1,3
       IFP(L)=1
       IF(ICOR(L,K).GT.ID(K+3).OR.ICOR(L,K).LT.ID(K)) IFP(L)=2
     8 VALF(L)=99999.9
       IF(IFP(L).EQ.1) VALF(L)=FUNCT(ICOR,L)
     6 CONTINUE
       MUM=10000
       DO 7 I=1,8
       IF(MUM.LT.VALF(I)) GO TO 7
       MUM=VALF(I)
       IC=I
     7 CONTINUE
       NEWC(1)=ICOR(IC,1)
       NEWC(2)=ICOR(IC,2)
       NEWC(3)=ICOR(IC,3)
       FNEW=VALF(IC)
```

```
      RETURN
      END
      FUNCTION FUNCT(NCOR,L)
      INTEGER NCOR(8,3),L
      FUNCT=9.0-8.0*NCOR(L,1)-6.0*NCOR(L,2)-4.0*NCOR(L,3)
     1+2.0*NCOR(L,1)**2+2.0*NCOR(L,2)**2+NCOR(L,3)**
     12+2.0*NCOR(L,1)*NCOR(L,2)+2.0*NCOR(L,1)*NCOR(L,3)
      RETURN
      END
      SUBROUTINE UNFORM(D,X,K)
C     ****************************************
C     TO GENERATE RANDOM POINTS IN THE DOMAIN
C     ****************************************
      REAL D(6)
      CALL RAND(RNI)
      X=D(K)+(D(K+3)-D(K))*RNI
      RETURN
      END
      SUBROUTINE RAND(R)
      REAL R,S,XM
      INTEGER IS
      DATA S,XM/.23978,823.53478/
      S=S*XM
      IS=S
      R=S-IS
      S=R+.8234617
      RETURN
      END
*EOR
```

Bibliography

1. H. Kuhn and A. Tucker. "Nonlinear Programming," in Proc. 2nd Berkley Symp. on Mathematical Statistics and Probability, J. Neyman, ed., Univ. of California Press, 1951.

2. A. Waren and L. Lasdon. "The status of Nonlinear Programming Software," Operations Research, 27, 431 (1979).

3. J. Kuester and J. Mize. Optimization Techniques with FORTRAN. New York: McGraw-Hill, 1973.

4. Leighton, Walter. "A Substitute for the Picone Formula," Bulletin of the American Mathematical Society, 55: 325-328 (1949).

5. Jon Mathews and R. L. Walker. Mathematical Methods of Physics. New York: W. A. Benjamin, Inc, 1965.

6. Komkov, Vadim. "A Generalization of Leighton's Variational Theorem," Journal of Applicable Analysis: 377-383 (1972).

7. Mahfoud, W, E. "Some Properties of Solutions of $(r(t)c(x)x')' + a(t)f(x) = 0$," Siam Journal on Mathematical Analysis, Vol. 10 No. 1: 49-54 (1979).

8. Kreyszig, J. Advanced Engineering Mathematics. New York: Wiley and Sons, Inc, 1962.

9. Rao and Mitra. Generalized Inverse of Matrices and its Applications. New York: Wiley and Sons, Inc,1971.

10. Doma, Mohamed A. Generalized Inverse of Matrices. Ms. thesis, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, December 1983.

11. Ortega and Rheinboldt. Iterative Solution of Nonlinear Equations in Several Variables. New York: Academic Press, 1970.

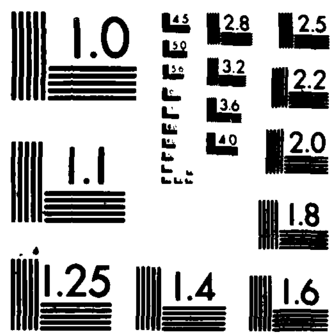
12. Ben-Israel Adi and Thomas N. E. Generalized Inverses: Theory and Applications. New York: Wiley and Sons, 1974.

13. Jones, John, Jr. "Solutions of Certain Matrix Equations," Proceedings of the American Mathematical Society, 31: 333-339 (February 1972).

14. Peter, Lancaster. Theory of Matrices. New York: Academic Press, 1969.

AD-A141 097 SPECIAL NONLINEAR OPTIMIZATION TECHNIQUES(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING A M RAGAB DEC 83 AFIT/MA/GOR/83D-5 2/2

UNCLASSIFIED F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

15. Borah, B, N and Chew, J, F. Some Common-Sense Optimization Techniques for Non-Differentiable Functions of Several Variables, June 2, 1983. Contract DAAG29-80-G-0004. Mathematics and Computer Science Dept, North Carolina A&T State University.

16. Brooks, S. H. "A Discussion of Random Methods for Seeking Maxima," Operations Research, 6: 244-251 (1958).

17. Becker, R. W. and Lago, G. V. "A Global Optimization Algorithm," Proceeding of the Eigth Allerton Conference on Circuits and System Theory, (1970).

18. Price, W. L. "A Controlled Random Search Procedure for Global Optimization," Computer Journal, Vol.20, No.4 (1977).

19. Nelder, J. A. and Mead, R. "A Simplex Method for Function Minimization," Computer Journal, Vol.7: 308-313 (1965).

VITA

Colonel Ali M. Ragab was born on 7 July 1946 in Egypt. He attended the Military Technical College in Cairo from which he received the degree of Bachelor of Science in Electrical Engineering in April 1968. During his service as a radar engineer in the Egyptian Army, he was awarded the following degrees:

(1) Diploma of Graduate Studies in Operations Research, May 1978, Institute of Statistical Studies and Research, Cairo University, Cairo

(2) Diploma of Graduate Studies in Industrial Business Management, May 1979, College of Commerce, Ain-Shams University, Cairo.

(3) Diploma of Graduate Studies in Antennas and Propagation, October 1980, Military Technical College, Cairo.

(4) M.Sc. in Operations Research, May 1982, Institute of Statistical Studies and Research, Cairo University, Cairo.

During the war of 1973 in the Middle East, he was rewarded the Republican Medal, First Class. He entered the School of Engineering, Air Force Institute of Technology, in July 1982.

Permanent address: 2 Block 81, Region 8
Nasr City, Cairo.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION

1b. RESTRICTIVE MARKINGS
AI41097

2a. SECURITY CLASSIFICATION AUTHORITY
Unclassified

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/MA/GOR/83-3

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State and ZIP Code)*
Air Force Institute of Technology,
Wright Patterson AFB,OH,45433

7b. ADDRESS *(City, State and ZIP Code)*

8a. NAME OF FUNDING/SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State and ZIP Code)*

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE *(Include Security Classification)*
See Block 19

12. PERSONAL AUTHOR(S)
Ali M. Ragab

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Yr., Mo., Day)*
Dec 1983

15. PAGE COUNT
98

16. SUPPLEMENTARY NOTATION

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)

17. COSATI CODES

FIELD	GROUP	SUB. GR.
12	01	
12	01	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Nonlinear Programming -Differential Equations

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Title: Special Nonlinear Optimization Techniques

Committee Chairman: Dr.John Jones,Jr

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL
Dr. John Jones,Jr

22b. TELEPHONE NUMBER *(Include Area Code)*
(513)255-3636

22c. OFFICE SYMBOL
AFIT/ENC

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

Thesis thesis extends the work of Leighton and Jones which takes functions that satisfy special types of differential equations and determines an interval on which the functions either have zeros or attain bounded values. Theorems for locating zeros are proved for functions of a single variable and functions of several variables with illustrative examples. Theapplications of matrix equations to constrained optimization problems are described. An algorithm for random search technique for the general optimization problem is presented with a FORTRAN V program and test problems.

END

FILMED

6-84

DTIC