RATE OF CONVERGENCE IN NONLINEAR PROGRAMMING

by

Vinod Chachra

Thesis submitted to the Graduate Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Industrial Engineering and Operations Research

APPROVED:

P. M. Ghare

Dr. P. M. Ghare, Chairman

M. H. Agee

Dr. M. H. Agee

Dr. C. J. Bellas

W. J. Fabrycky

Dr. W. J. Fabrycky

Dr. J. W. Schmidt

May, 1972

Blacksburg, Virginia

ACKNOWLEDGEMENTS

The author is grateful for this opportunity to express his appreciation to the following individuals for their aid and encouragement in the completion of this study.

Dr. P. M. Ghare, the author's major advisor, for providing invaluable guidance and advice during all stages of the study.

Dr. J. W. Schmidt, member of his graduate committee, for his encouragement and invaluable aid in the mathematical developments in this study.

Dr. M. H. Agee, member of his graduate committee and Head of Industrial Engineering and Operations Research Department, for his advice and valuable suggestions in the writing and editing of this manuscript.

The other members of his graduate committee, Dr. C. J. Bellas and Dr. W. J. Fabrycky, for their assistance and constructive criticisms.

Mr. H. H. Wiss and Dr. R. E. Taylor, close friends of the author, who were constantly available for advice, encouragement and council.

The author's parents, Mr. and Mrs. Nandlal Chachra for their patience, encouragement and support.

Mrs. Gail P. Metzger for preparing the original draft of the manuscript.

Mrs. Emily Holland for preparing the final draft of the manuscript and excellent typing.

TABLE OF CONTENTS

Page

CHAPTER

I. INTRODUCTION 1

1. Importance of Study 2

2. Specific Objectives of the Research 4

3. Approach 6

4. Summary of Previous Work 6

II. SOME BASIC CONCEPTS IN CONVERGENCE 7

III. NEWTON-RAPHSON PROCEDURE 13

1. Newton-Raphson Method for Scalar Functions 14

2. Higher Order Method for Scalar Functions 15

3. Newton-Raphson Method for Multivariate Functions 19

4. Higher Order Procedure for Multivariate Functions 23

IV. THE SIMPLEX ALGORITHM 25

V. UNCONSTRAINED OPTIMIZATION BY GRADIENT METHODS 34

1. Gradient Method 36

2. General Directional Method 43

VI. ROSEN'S METHOD 47

VII. CONCLUSION 53

1. Summary 53

2. Suggestions for Further Study 54

REFERENCES 56

VITA 58

I. INTRODUCTION

The Nonlinear programming problem [NLP] can be stated as follows: Let $f_i(X) = f_i\ (X_1, X_2, \ldots, X_n)$ be a continuous real valued function of n variables for each $i = 0, 1, \ldots, m$. The problem is to find, within a reasonable approximation a value of M defined by:

$$M = \text{Inf}\ \{f(X) : X \varepsilon S\},$$

where set S is the constraint set

$$S = \{X : h_i(X) \leq 0,\ i > 0\}\ .$$

Most of the work done in the area of NLP has been restricted to determining the necessary and sufficient conditions for optimality under various assumptions about the functions h_i, $i = 0, 1, \ldots, m$. The practical importance of the problem has led to the development of numerous algorithms for solving some subset of the problem stated above. However, relatively little work has been done in "the study of the process of solution, that is, of the nature and rate of convergence of a particular procedure for solution applied to a particular type of problem". [17]

Two questions regarding convergence may be asked of each algorithm, namely,

a) Does the algorithm converge to the optimal?

b) How fast does it converge?

Most designers of algorithms provide a proof for its convergence. Some merely state that no proof is available, however, they claim that in all practical applications it is found to converge [7]. Zangwill [18] has recently presented a comprehensive theory of convergence. Within the framework of this theory the proofs of convergence have been greatly simplified. However, even Zangwill, in referring to the rate of convergence, merely says that one procedure is 'better' than the other. There are but a few algorithms for which the task of determining the rate of convergence has been accomplished. It is this specific question on the rate of convergence that this research will address itself to.

1. Importance of Study

Numerous algorithms have been designed for the solution of specific nonlinear programming problems. Computer routines are available for many of these algorithms. In deciding which algorithm to use for a given set of problems, consideration must be given to the following aspects:

a) Availability of packaged program

b) Familarity with algorithm

c) Amount of computation per iteration

d) Convergence of algorithm

e) Rate of convergence of algorithm

f) Accuracy of the computation.

If it is required to solve only a very few problems, then the choice invariably is to select the algorithm for which a packaged program is available. If several packaged programs are available, then the most familiar or convenient to use algorithm is selected. However, if an algorithm has to be used repeatedly, as in a production control type situation, then importance is given to the convergence and computational aspects of the algorithm. Even here, there may be situations where the accuracy of the result is very important. In such cases the user may be willing to preserve accuracy at the cost of increased computations or slow convergence. Again, there are algorithms which achieve a better rate of convergence as a result of increased computations per iteration. The Newton-Raphson procedure, for example, has a poorer convergence rate than the related higher order procedure (both discussed in Chapter III), but the latter requires more computation. If, for the specific problems, it is found that the rate of convergence is not appreciably better for the higher order procedure, then its additional computational effort may not be justified. Barring these exceptions, an algorithm with a better rate of convergence is always preferred.

Thus, the rate of convergence of an algorithm is an important and useful measure of its performance. Prior knowledge of the rate of convergence of the algorithms permits a comparison of their performance. It is clear that in order to make a comparison amongst algorithms, the rates of convergence for all of them must be made available. This research is an effort towards that end.

2. Specific Objectives of the Research

The NLP has the following three principal solution procedures:

a) Solution procedures that reduce the constrained problem to an equivalent problem without constraints. The methods of Lagrange, Fiacco and McCormick, and Hildreth serve as examples.

b) Methods that have the solution point moving from one vertex to another along an edge of the constraint set. These may result from approximating the nonlinear problem by a sequence of linear problems. The algorithms by Wolfe, and Beale fall into this category.

c) Methods where the move is in the interior of the feasible region, i.e., the solution procedure does not necessarily move along the edge. Examples of this type are algorithms by Zoutendijk, Rosen and Greenstadt.

It is proposed to study the rate of convergence of some of the algorithms in this classification by types. The objective is to obtain expressions from which the rate of convergence for the algorithm may be computed. In choosing the algorithm for study, effort was made to ensure that at least one algorithm came from each classification above. The specific algorithms studied are listed below with a brief explanation for its choice.

1) Newton-Raphson method: A NLP problem can readily be reduced to a system of nonlinear equations by the method of Lagrange or by some penalty function type method. The Newton Raphson procedure provides an iterative scheme for solving the system of nonlinear equations. It is proposed to study the rate of convergence of the Newton-Raphson

procedure for both scalar and multivariate functions. A higher order procedure which is a natural consequence of the Taylor series expansion is also studied. This study is presented in Chapter III.

2) Simplex Method: Even though the simplex method is used only for linear problems, its study is included since it forms the root of many nonlinear algorithms. This is a type (b) algorithm, and its rate of convergence is presented in Chapter IV. [The original objective was to find an explicit expression for the rate of convergence. Failing to find satisfactory results, the objective was reduced to finding an expression for the improvement in the objective function in terms of the original problem, rather than intermediate computations.]

3) Gradient Method: Since gradient techniques are so basic in nonlinear programming and form the basis for many directional procedures, its study as a type (c) algorithm was considered desirable. Chapter V deals with the performance of the optimal gradient method for quadratic programming problems and the results for the optimal gradient method are generalized for any directional method.

4) Rosen's Method: This is a type (c) algorithm depending on the gradient method. The rate of convergence for Rosen's algorithm, under very strict assumptions about the computational procedure, are available in the literature [15]. In this study an effort was made to find better estimates of the rate of convergence under less stringent assumptions. The result of the study is presented in Chapter VI.

3. Approach

There is no unified or single approach in determining the rate of convergence and thus each algorithm must be treated on its own merits. The idea is to try and find the lower and upper bounds for the ratio of $f(X_{k+1})$ to $f(X_k)$ in terms of the iteration number k or the data available at the first iteration ($f(X_k)$ is the value of the objective function at the k^{th} iteration). In view of this, a common starting point is to express $f(X_{k+1})$ in terms of $f(X_k)$. Beyond this, there is no similarity between procedures and each algorithm has to be treated according to its computational features.

4. Summary of Previous Work

In the area of rate of convergence of NLP algorithms Kantorovich [9] first established that the gradient method converged at the rate of a geometric progression for quadratic functions. Akaike [1] showed that the best convergence for the gradient method for quadratic functions was of the same form as given by Kantorovich for the worst convergence. Cannon and Cullum [3] showed that the slowest and the fastest convergence for the Frank-Wolfe algorithm had the same form. Similar results are presented by Wolfe [17] for Kelly's cutting plane algorithm and for the decomposition algorithm. Rosen [15], under very strict assumptions of the computational procedure, presented a lower bound on the rate of convergence of his method. These are the only algorithms for which the task has been accomplished [17].

II. SOME BASIC CONCEPTS IN CONVERGENCE

This chapter is devoted to defining certain terms and theorems which will be found useful in subsequent chapters. The material of this chapter is particularly useful for the developments presented in Chapter III. References [2], [5], [6], [9], [13] and [17] were used in compiling the information presented herein.

Convergence theorems must deal with the following two questions regarding convergence:

i) Does a procedure converge? If so, under what conditions?

ii) Does it converge to the correct solution?

Theorem 2.1 on page 8 deals with the second question regarding convergence. This theorem defines a condition which assures that an iterative procedure, if convergent, will converge to the correct solution. The iterative procedures in Chapter III meet the requirements of this theorem.

Theorem 2.2 on page 9 addresses itself to the second question. It presents a set of sufficient conditions for the convergence of an iterative procedure. The discussion following the theorem demonstrates a simple test to establish convergence. In essence, in order to establish convergence, one need only show that the derivative of an equivalent form is less than unity in the neighborhood of the solution and that the starting point lies in this neighborhood. This particular result is used in Chapter III to establish the convergence for scalar functions. Theorem 2.3 on page 12 is a multivariate generalization of

Theorem 2.2 and is used for the multivariate cases in Chapter III. The following terms need to be defined before the statement of these theorems:

Definition 2.1 -- Convergent Sequence: A sequence is said to be convergent if it has a limit, otherwise it is divergent.

Definition 2.2 -- Rate of Convergence: The rate of convergence is some quantity that permits one to predict, given the worst situations, how hard it will be to solve a problem. It is therefore a quantity that provides a bound for the slowest convergence.

Definition 2.3 -- Order of Convergence: If a sequence produced by an algorithm asymtotically satisfies a relation of the form

$$|f(x_{i+1}) - M| = C|f(x_i) - M|^p,$$

where C and M are constants, then p is called the order of convergence.

Definition 2.4 -- Equivalent Forms: Two equations are said to be equivalent if they have the same roots. If $f(x) = 0$ and $x = g(x)$ are equivalent forms, then the second relation suggests an iterative procedure

$$x_{i+1} = g(x_i), \quad \text{for} \quad i = 0, 1, 2, \ldots \tag{2.1}$$

Theorem 2.1

$$\text{If} \quad f(x) = 0 \tag{2.2}$$

$$\text{and} \qquad x = g(x) \tag{2.3}$$

are equivalent forms, and if the sequence represented by (2.1) converges to a limit L, then $x = L$ is a solution to (2.2).

Proof: Taking the limit of (2.1) as $i \rightarrow \infty$ gives $L = g(X)$, which is a solution to (2.3) and, by definition, each solution to (2.3) is also a solution to (2.2).

Definition 2.5 -- ε-neighborhood: The "ε-neighborhood" of L, denoted by $N(L,\varepsilon)$, where $\varepsilon > 0$, is defined by

$$N(L,\varepsilon) = \{x : |X - L| < \varepsilon\} .$$

Theorem 2.2

If $f(x) = 0$ and $x = g(x)$ are equivalent forms and $g(x)$ is analytic at $x = L$, where L is the solution to these equations,

and if there exist two positive numbers ε and K such that $K < 1$ for which

$$|g(x) - g(L)| < K|x - L| \quad \text{for all } x \varepsilon N(L,\varepsilon), \tag{2.4}$$

and if x_0 is a point in $N(L,\varepsilon)$,

then the sequence $x_0, x_1, x_2, \ldots$, which is defined by $x_{i+1} = g(x_i)$, will converge to L.

The proof of this theorem is presented by Macon [13]. A more important consequence, which is used in Chapter III, is now illustrated. Consider a sequence $x_0, x_1, x_2, \ldots$ Then, in terms of the previous notation:

$$L - x_{i+1} = g(L) - g(x_i) \; .$$

It is known from the mean value theorem that there exists a value e_i lying between x_i and L, such that

$$L - x_{i+1} = (L - x_i)g'(e_i) \; .$$

But as $i \to \infty$, then $x_i \to L$ and hence, $e_i \to L$. Thus as $i \to \infty$, then

$$(L - x_{i+1}) = (L - x_i)g'(L) \; .$$

We note that if $g'(L) < 1$, then the condition (2.4) of theorem 2.2 is satisfied for any K in the range $g'(L) \leq K < 1$. Thus, in order to establish convergence, one need only show that $g'(L) < 1$, and that the starting point is in the required neighborhood.

The definition of order of convergence given earlier is difficult to use. A more convenient definition may now be obtained by a Taylor series expansion of g(x) as follows:

$$g(x) = g(L) + (x - L)g'(L) + \ldots + \frac{(x - L)^{n-1}g^{n-1}(L)}{(n - 1)!}$$

$$+ \frac{(x - L)^n g^n(L)}{n!}$$

or,

$$g(x) - g(L) = (x - L)g'(L) + \ldots + \frac{(x - L)^{n-1}g^{n-1}(L)}{(n - 1)!} \; .$$

If we now consider only the first non-zero term on the right hand side, then definition 2.3 is satisfied. Hence the order of convergence may be redefined as

If $g'(L) = g^2(L) = \ldots = g^{(p-1)}(L) = 0$ and

$g^p(L) \neq 0$, then the convergence is said to be of order p.

Theorem 2.2 and its subsequent discussion deals only with a single variable case. The entire discussion can be extended to include the n-variable case.

Define $F(X) = 0$ to be the system of nonlinear equations to which a solution is desired and $\Phi(X)$ to be the matrix of first partials of $F(X)$. Also, define $X = G(X)$ to be an equivalent form of $F(X) = 0$. That is,

$$F(X) = \begin{bmatrix} f_1(X) \\ f_2(X) \\ \cdot \\ \cdot \\ \cdot \\ f_n(X) \end{bmatrix}, \qquad \Phi(X) = \begin{bmatrix} \nabla f_1(X) \\ \nabla f_2(X) \\ \cdot \\ \cdot \\ \cdot \\ \nabla f_n(X) \end{bmatrix},$$

$$G(X) = \begin{bmatrix} g_1(X) \\ g_2(X) \\ \cdot \\ \cdot \\ \cdot \\ g_n(X) \end{bmatrix}, \quad \text{and} \quad \Theta(X) = \begin{bmatrix} \nabla g_1(X) \\ \nabla g_2(X) \\ \cdot \\ \cdot \\ \cdot \\ \nabla g_n(X) \end{bmatrix}.$$

Define $N(X)$ as the vector norm of X and $N(A)$ the corresponding matrix norm. Then we may state the following theorem:

Theorem 2.3

If $F(X) = 0$ and $X = G(X)$ are equivalent forms, and if,

1) $G(X)$ is defined and has continuous first derivatives for X in a convex domain R,

2) $N(\Theta(X)) \leq K < 1$, for all $X \varepsilon R$,

3) $X_0 \varepsilon R$, and the sphere $\{X | N(X - X_0) \leq \dfrac{N(G(X_0) - X_0)}{(1 - K)}\} \subset R$,

then the iterative process defined by $X = G(X)$ converges to the solution of $F(X) = 0$.

Proof: The proof may be found in John [9].

III. NEWTON-RAPHSON PROCEDURE

The Newton-Raphson method is useful in solving a system of nonlinear equations. In NLP systems of nonlinear equations may arise in the optimization of unconstrained problems. An example is the euclidean facility location problem for single or multiple facilities. Also, the Lagrange method for constrained optimization, reduces the constrained problem to an equivalent problem without constraints. Then, taking the partial derivatives of the equivalent problem with respect to all the variables and setting them equal to zero again results in a system of nonlinear equations. There are thus a wide variety of problems that lend themselves to solution by the Newton-Raphson procedure.

The Newton-Raphson method has convergence of order two. However, for a single variable problem this procedure is applicable only to functions which have a non-zero first derivative arbitrarily close to the solution. [In a n-variable problem, the Jacobian determinant of the function should be non-zero.] In order to use the Newton-Raphson procedure, only the first partial derivatives of the nonlinear system of equations need to be computed. The rate of convergence may be improved by calculating higher order derivatives. However, increased rate of convergence comes at the cost of extra computations per iteration. A higher order method which is closely related to the Newton-Raphson method is suggested in section 2. The rate of convergence of both the Newton-Raphson procedure and a related higher order procedure are studied in this dissertation and no attempt is made to determine if the increased rate of convergence of the higher order procedure in fact justifies the increased computations.

The convergence of both procedures is established from the results of theorems 2.2 and 2.3. Thus, it is necessary for both procedures that the starting point lie within the circle in which the function defining the iterative procedure satisfies the conditions of the above theorems.

1. Newton-Raphson Method for Scalar Functions

The contents of this section follow the presentation in Macon [13]. The Newton-Raphson method is used to find the solution x^*, to $f(x) = 0$. The procedure defines an equivalent form as

$$g(x) = x - \frac{f(x)}{f'(x)} , \quad \text{if } f'(x) \neq 0 . \tag{3.1}$$

Differentiating (3.1) gives

$$g'(x) = 1 - \frac{f'(x)}{f'(x)} + \frac{f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2} .$$

Since $\quad f(x^*) = 0 ,$ $\hfill (3.2)$

then $\quad g'(x^*) = 0 .$ $\hfill (3.3)$

Referring to theorem 2.2 and the contents of Chapter II, equation (3.3) establishes the convergence and indicates that the procedure is at least of order two.

Taking the second derivative results in

$$g''(x) = \frac{f(x)[f'(x)]^2 f^3(x) + [f'(x)]^3 f''(x) - 2f(x)f'(x)[f''(x)]^2}{[f'(x)]^4}$$

We note that at $x = x^*$, $f(x^*) = 0$.

Hence, $$g''(x^*) = \frac{f''(x^*)}{f'(x^*)} \quad . \tag{3.4}$$

Equation (3.4) indicates that the procedure is of order two and that the error may be approximated by

$$(x_{i+1} - x^*) = \frac{(x_i - x^*)^2 f''(x^*)}{2f'(x^*)} = \frac{1}{2}(x_i - x^*)^2 g''(x^*) \quad . \tag{3.5}$$

Thus the rate of convergence may be measured by $\left|\frac{g''(x^*)}{2!}\right|$ and we observe that the smaller this value, the faster the convergence.

2. Higher Order Method for Scalar Functions

A higher order method, closely related to Newton-Raphson method is derived from the Taylor series expansion for a function. The Taylor series expansion for $f(x_i + h)$ is

$$f(x_i + h) = f(x_i) + hf'(x_i) + \frac{h^2}{2} f''(x_i) + \ldots$$

At the solution point, $f(x_i + h) = 0$, which yields

$$h = \frac{- f'(x_i) \pm \sqrt{[f'(x_i)]^2 - 2f(x_i)f''(x_i)}}{f''(x_i)} . \tag{3.6}$$

Equation (3.6) suggests the iterative procedure shown in equation (3.7). Taking the positive sign, the function g(x) may be defined as:

$$g(x) = x + \frac{\{- f'(x) + \sqrt{[f'(x)]^2 - 2f(x)f''(x)\}}}{f''(x)} . \tag{3.7}$$

Clearly the procedure requires the computation of second-order derivatives to define the new point. Now, letting R equal the quantity under the radical and proceeding as in section 1, gives

$$g'(x) = 1 + \quad \frac{- f''(x)}{f''(x)} + \frac{1}{2} \cdot \frac{2f'(x)f''(x) - 2f''(x)f''(x)}{\sqrt{R}\, f''(x)}$$

$$+ \frac{1}{2} \cdot - \frac{2f(x)f^3(x)}{\sqrt{R}\, f''(x)}$$

$$- \quad \frac{(- f'(x) + \sqrt{R})}{[f''(x)]^2} \cdot f^3(x) ,$$

or

$$g'(x) = - \frac{f(x)f^3(x)}{\sqrt{R}\, f''(x)} - \quad \frac{(- f'(x) + \sqrt{R})f^3(x)}{[f''(x)]^2} \quad . \tag{3.8}$$

Since $f(x^*) = 0$, then

$$g'(x^*) = - \frac{(- f'(x) + f'(x))f^3(x)}{[f''(x)]^2} = 0 .$$

Again, by theorem 2.2 we conclude that the higher order procedure converges and that the order of convergence is at least two. To obtain the exact order, the next higher derivative needs to be calculated.

Thus,

$$g''(x) = - \frac{f'(x)f^3(x)}{\sqrt{R}\ f''(x)} + \frac{f(x)f^4(x)}{\sqrt{R}\ f''(x)} - \frac{f(x)[f^3(x)]^2}{\sqrt{R}\ [f''(x)]^2}$$

$$+ \frac{[f(x)]^2[f^3(x)]^2}{f''(x)R^{3/2}} + (\sqrt{R} - f'(x)) \frac{f^4(x)}{[f''(x)]^2}$$

$$- \frac{2[f^3(x)]^2}{[f''(x)]^3} + \frac{f^3(x)}{[f''(x)]^2} - f''(x) + \frac{f(x)f^3(x)}{\sqrt{R}}$$

Again, at $x = x^*$, and $f(x^*) = 0$

gives $g''(x^*) = 0$.

Hence, the convergence is at least of order 3, and the next higher derivative needs to be calculated.

Noting that $f(x^*) = 0$ and $\frac{d}{dx}\left[\frac{1}{\sqrt{R}}\right]_{x=x^*} = 0$

and that, $\sqrt{R}\,\Big|_{x=x^*} = f'(x^*)$, it can be shown that

$$g^3(x^*) = \frac{f^3(x^*)}{f'(x^*)} \quad \text{when } f'(x^*) \neq 0 \,. \tag{3.9}$$

Equation 3.9 indicates that convergence is of order 3 and the rate of convergence may be measured by $\left|\frac{g^3(x^*)}{3!}\right|$ or $\left|\frac{f^3(x^*)}{f'(x^*)3!}\right|$, the smaller the value, the faster the convergence.

The higher order procedure discussed here results as a natural consequence of Taylor series expansion. There are several higher order procedures cited in literature. As far back as 1838 Chebyshev (p. 142 in [2]) proposed a method based on a representation of the inverse function of f(x). A method for obtaining higher order iterations from one or two iterations of the same order was proposed by Aitken (p. 148 in [2]). Thus, if $x = \phi_1(x)$ and $x = \phi_2(x)$ are two iterative schemes of the same order, a higher order procedure $\phi(x)$ is given by

$$\phi(x) = \frac{x\phi_1[\phi_2(x)] - \phi_1(x)\phi_2(x)}{x - \phi_1(x) - \phi_2(x) + \phi_1[\phi_2(x)]} \,.$$

3. Newton-Raphson Method for Multivariate Functions

The general multivariate case for Newton-Raphson method is rarely treated in literature and, to the best of the author's knowledge, this particular simplified approach has not been used. The computational details of the procedure may be found in Wilde and Beightler [16].

The development of the rate of convergence follows. Let $F(X) = 0$ be a system of nonlinear equations to which a solution X^* is desired. Define $F(X)$, $G(X)$, $\Phi(X)$, and $\Theta(X)$ as in Chapter II. Then, the iterative formula may be defined as

$$G(X) = X + \delta(X) , \tag{3.10}$$

where

$\delta(X)$ is a solution to

$$F(X) + \Phi(X)\delta(X) = 0 \quad \text{and} \quad |\Phi(X)| \neq 0 . \tag{3.11}$$

Differentiating 3.10 results in

$$\frac{\partial}{\partial x_i} G(X) = \frac{\partial}{\partial x_i} [X] + \frac{\partial}{\partial x_i} \delta(X) \qquad \text{for } i = 1, 2, \ldots, n \tag{3.12}$$

In view of theorem 2.3, in order to prove convergence, it is sufficient to show that

$$\text{Norm } [\Theta(X)] < 1 , \tag{3.13}$$

which is accomplished below.

In order to establish (3.13) the value of $\frac{\partial}{\partial x_i} \delta(X)$ must be calculated for use in equation (3.12). The value of $\frac{\partial}{\partial x_i} \delta(X)$ may be obtained by differentiating (3.11).

Thus,

$$\frac{\partial}{\partial x_i} F(X) + \Phi(X) \frac{\partial}{\partial x_i} \delta(X) + \frac{\partial}{\partial x_i} \Phi(X)\delta(X) = 0 \qquad (3.14)$$

$$\text{for } i = 1, 2, \ldots, n \ .$$

Noting that at $X = X^*$, $\delta(X^*) = 0$,

gives

$$\frac{\partial}{\partial x_i} F(X^*) + \Phi(X^*) \frac{\partial}{\partial x_i} \delta(X^*) = 0 \qquad \text{for } i = 1, 2, \ldots, n \qquad (3.15)$$

Equation (3.15) represents a system of n^2 equations which may be rewritten as:

$$\frac{\partial}{\partial x_i} f_k(X) + \nabla f_k(X) \frac{\partial}{\partial x_i} \delta(X) = 0 \ , \qquad \forall i, \forall k \quad \text{at} \quad X = X^*$$

or,

$$\frac{\partial}{\partial x_i} f_k(X) + \frac{\partial}{\partial x_i} f_k(X) \frac{\partial}{\partial x_i} \delta_i(X) + \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\partial}{\partial x_j} f_k(X) \frac{\partial}{\partial x_i} \delta_j(X) = 0$$

$\forall i, \forall k$ at $X = X^*$.

This last equation may conveniently be represented in matrix form as

$$\Phi(X) \begin{bmatrix} [1 + \frac{\partial}{\partial x_1} \delta_1(X)] & \frac{\partial}{\partial x_2} \delta_1(X) & \cdots & \frac{\partial}{\partial x_n} \delta_1(X) \\ \frac{\partial}{\partial x_1} \delta_2(X) & [1 + \frac{\partial}{\partial x_2} \delta_2(X)] & \cdots & \\ \vdots & \vdots & & \vdots \\ \frac{\partial}{\partial x_1} \delta_n(X) & \frac{\partial}{\partial x_2} \delta_n(X) & \cdots & [1 + \frac{\partial}{\partial x_n} \delta_n(X)] \end{bmatrix} = [0]$$

Since $\Phi(X)$ by assumption is non-zero, it implies that,

$$\left.\begin{array}{ll} \frac{\partial}{\partial x_i} \delta_i(X) = -1 & \forall_i \\ \frac{\partial}{\partial x_i} \delta_j(X) = 0 & \forall_i, \forall_{j \neq i} \ . \end{array}\right\} \qquad (3.16)$$

Substitution of equation (3.16) in (3.12) establishes that

$$\frac{\partial}{\partial x_i} G(X^*) = 0 \qquad \forall_i \ .$$

Hence, Norm $[\Theta(X)] < 1$ and the procedure converges with an order of at least two. To establish the exact order, the higher order partial derivative is computed next. Differentiating equation (3.14), and noting that $\delta(X^*) = 0$, we obtain

$$\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}F(X^*) + \frac{\partial}{\partial x_j}\Phi(X^*)\frac{\partial}{\partial x_i}\delta(X^*) + \Phi(X^*)\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}\delta(X^*)$$

$$+ \frac{\partial}{\partial x_i}\Phi(X^*)\frac{\partial}{\partial x_j}\delta(X^*) = 0 . \tag{3.17}$$

Also equation (3.12) indicates that

$$\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}G(X) = \frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}\delta(X) . \tag{3.18}$$

From equation (3.17) and (3.18) we can conclude that $\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}G(X)$ is other than zero and hence the procedure is in fact of order two. Again, the error may be approximated by a Taylor series expansion which yields:

$$(X_{k+1} - X^*) = \frac{1}{2!}D^2G(X)\Big|_{X=X^*}$$

where D is an operator defined as

$$D = \sum_{i=1}^{n} E_i(X_k)\frac{\partial}{\partial x_i} ,$$

and $E_i(X_k)$ is the ith component of the error vector at the kth iteration. This last equation gives an estimate for the rate of convergence of the procedure.

4. Higher Order Procedure for Multivariate Functions

The higher order procedure defined for scalar functions in section 2 may be generalized for multivariate functions. The iterative process for an n-vector X is given by

$$G(X) = X + \delta(X) \ , \tag{3.19}$$

where $\delta(X)$ is the solution to

$$F(X) + \sum_{i=1}^{2} \frac{1}{i!} D^i F(X) = 0 \ . \tag{3.20}$$

The operator D is

$$D = \sum_{j=1}^{n} \delta_j(X) \frac{\partial}{\partial x_j}$$

and $\delta_i(X)$ is the i^{th} component of $\delta(X)$.

Whereas for scalar functions a higher order procedure may be found useful, in the case of multivariate functions it can be a formidable task to get a solution to (3.20), thereby reducing its usefulness. It is obvious that the computations required to solve equation (3.20) will be significantly greater than that required to solve equation (3.11) for the Newton Raphson procedure. The solution of equation (3.11) in itself is no easy task. It requires the computation of the inverse of the matrix $\Phi(X)$, a task that increases rapidly in magnitude with the size of the matrix. The computations of the inverse also propogate round off errors thereby preventing the solution to be reached with a desired degree of accuracy [2]. Since the simpler equation (3.11) presents such undesirable traits for large size problems, it would indeed be unwise to even attempt to solve equation (3.20). In view of this, it was decided not to pursue the rate of convergence for this method. For multivariate functions the gradient method offers a useful procedure for the solution of a system of equations, and gives the required solution with a desired degree of accuracy. Its rate of convergence for solution to linear systems is discussed in Chapter IV.

IV. THE SIMPLEX ALGORITHM

The simplex algorithm has been used extensively for the solution of linear problems with linear constraints. Further, there are solution procedures for the nonlinear problem, like those by Beale, Barkin and Dorfman, Frank and Wolfe, which depend heavily on the simplex algorithm. It has been established that the simplex method converges in a finite number of steps, but it is more difficult to establish a rate of convergence for this algorithm. Klee [11] has obtained estimates for the maximum number of iterations for problems of a given size. He found that the rule which, at each step, maximizes the gradient in the space of non-basic variables requires, for an n variable m equality constraint problem, as many as $m(n - m - 1) + 1$ iterations. By contrast, a second rule which, at each step, tries to maximize the improvement in the objective function requires at most m iterations if $m = n - 2$, at most $[3m/2]$ if $m = n - 3$, and $2m - 1$ if $m \leq n - 4$. However, these estimates have not been definitely established as the maximum in question.

The rate of convergence of the simplex algorithm may also be estimated by the improvement in the objective function between successive iterations. The aim of the development in this chapter is to find an expression for the improvement in the objective function. The usefulness of the expression is increased if it is in terms of the original problem rather than intermediate computations. The final expression obtained in equation (4.14) is in terms of the original problem. In the paragraphs to follow, the linear programming problem is stated, a set of notations is defined, and in terms of these notations the development necessary to arrive at equation (4.14) is presented.

The linear programming problem may be stated as:

$$\begin{aligned} &\text{Minimize} \quad f(X) = (C,X) \\ &\text{subject to} \quad AX = b\ . \end{aligned} \tag{4.1}$$

The computations of the simplex algorithm are first expressed in terms of the product form of inverse. The following notation is used:

B - the present basis

N - the columns of A not included in B

$\overline{B}$, $\overline{N}$ - corresponding matrices for the new basis.

A, X and C can be expressed in terms of the basis as

$A = (B|N) = (\overline{B}|\overline{N})$

$X = (X_B|X_N)$, where X_B are the variables associated with the basis B

$C = (C_B|C_N)$, where C_B are the cost-coefficients associated with the basis B and C_N the cost-coefficients not associated with the basis B

C_s = the cost-coefficient associated with column s

Also define

r = the index of the departing variable

s = the index of the entering variable

P_j = the column of A associated with variable j

β_i = the row of B^{-1} associated with variable i

$\bar{a}_{..}$ = the elements of A during intermediate computations, and

$$B^{-1}P_s = \begin{bmatrix} \bar{a}_{1s} \\ \bar{a}_{2s} \\ \vdots \\ \bar{a}_{ms} \end{bmatrix} . \tag{4.2}$$

$$K_i = -\bar{a}_{is}/\bar{a}_{rs} , \quad i \neq r \tag{4.3}$$

$$K_r = 1/\bar{a}_{rs}$$

$$\bar{K} = \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ (K_i - 1) \\ \vdots \\ K_m \end{bmatrix} \tag{4.4}$$

and $f(X_B)$ = the value of the objective function when B is the basis.

In the development that follows the objective is to find an expression for the improvement in the objective function between successive iterations, i.e., an expression for $f(X_B) - f(X_{\bar{B}})$.

$$BX_B + NX_N = b \tag{4.5}$$

$$f(X_B) = (C_B, X_B) + (C_N, X_N) \tag{4.6}$$

$$= (C_B, X_B) \quad \text{as} \quad X_N = 0 \ . \tag{4.7}$$

Also,

$$C_{\overline{B}} = C_B + C_T \ , \quad \text{where } C_T = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ C_s - C_r \\ 0 \\ 0 \end{bmatrix} \leftarrow r^{th} \text{ row} \tag{4.8}$$

and

$\overline{K}\beta_r$ is an mxm square matrix.

It is possible to express the new inverse in the terms of the old inverse as follows:

$$\overline{B}^{\,-1} = [B^{-1} + \overline{K}\beta_r] \tag{4.9}$$

also,

$$f(X_B) = (C_B, X_B) = C_B^T \, B^{-1} b \ ,$$

and

$$f(X_{\overline{B}}) = (C_{\overline{B}}, X_{\overline{B}}) = C_{\overline{B}}^T \, \overline{B}^{\,-1} b \ .$$

Then,

$$f(X_B) - f(X_{\overline{B}}) = (C_B^T\, B^{-1}b - C_{\overline{B}}^T\, \overline{B}^{\,-1}b)$$

Using (4.8) and (4.9) gives

$$f(X_B) - f(X_{\overline{B}}) = [C_B^T\, B^{-1}b - (C_B + C_T)^T\, (B^{-1} + \overline{K}\beta_r)b]$$

$$= - C_B^T\, (\overline{K}\beta_r)b - C_T^T\, (B^{-1} + \overline{K}\beta_r)b\ . \qquad (4.10)$$

Now, using (4.2), (4.3) and (4.4) gives

$$C_B^T\, (\overline{K}\beta_r)b = \frac{C_B^T\, B^{-1}P_s\beta_r b}{-\bar{a}_{rs}} + \frac{C_r}{\bar{a}_{rs}}\, \beta_r b\ , \qquad (4.11)$$

and

$$C_T^T\, (B^{-1} + \overline{K}\beta_r)b = (C_s - C_r)\, \beta_r b + (C_s - C_r)(\frac{1}{\bar{a}_{rs}} - 1)\beta_r b$$

$$= \frac{(C_s - C_r)}{\bar{a}_{rs}}\, \beta_r b\ . \qquad (4.12)$$

Combining equations (4.10), (4.11) and (4.12) results in

$$f(X_B) - f(X_{\overline{B}}) = \frac{[C_B^T\, B^{-1}P_s - C_s]\beta_r b}{\bar{a}_{rs}}\ .$$

Let v_s be the value at which the variable s enters the solution, then

$$v_s = \frac{\beta_r b}{\bar{a}_{rs}}$$

Hence,

$$f(X_B) - f(X_{\overline{B}}) = [C_s - C_B^T B^{-1} P_s]\ v_s \qquad (4.13)$$

Equation (4.13) describes the improvement in the objective function between successive iterations of the simplex procedure. This improvement is in terms B^{-1}, the inverse of the basis matrix. The usefulness of (4.13) increases if it can be expressed in terms of the basis B. The matrix B in general is neither symmetric nor positive definite. Since eigenvalues of symmetric positive definite matrices are most convenient to work with, an effort is now made to find a symmetric positive definite matrix that may be substituted instead of B. However, the following lemma's first need to be established.

LEMMA 3.1 The eigenvalues of a positive definite symmetric matrix are real and positive.

Proof: The proof may be found in Hadley [8].

LEMMA 3.2 The eigenvalues of B^{-1} are the reciprocals of the eigenvalues of B.

Proof: Let B be a square matrix with its characteristic equation given by $f(\lambda) = 0$.

Now,

$$(B^{-1} - \lambda I) = B^{-1}[I - \lambda B] = - \lambda B^{-1}[B - \lambda^{-1}]$$

If $F(\lambda) = 0$ is the characteristic equation of B^{-1}

then $\quad F(\lambda) = |B^{-1} - \lambda I| = |(-\lambda B^{-1})(B - \lambda^{-1} I)|$

or $\quad F(\lambda) = (-\lambda)^n |B^{-1}| \; |B - \lambda^{-1} I|$

$$= (-\lambda)^n |B^{-1}| f(\tfrac{1}{\lambda}) = \frac{(-\lambda)^n}{|B|} f(\tfrac{1}{\lambda}) \; .$$

Since $|B| \neq 0$ and $(-\lambda)^n \neq 0$,

then $F(\lambda) = 0$ implies that $f(\frac{1}{\lambda}) = 0$.

This shows that the eigenvalues of B^{-1} are the reciprocals of the eigenvalues of B.

LEMMA 3.3

Define the norm of a vector X and a matrix A as follows:

$|X| = (X, X)^{1/2}$

$|\;|A|\;| = \sqrt{\lambda_n}$

where λ_n is the largest eigenvalue of $A^* A$ and A^* is the complex conjugate of A.

Then,

$|AX| \leq |\;|A|\;| \cdot |X| = \sqrt{\lambda_n}\;|X|$

Proof: The proof may be found in Fadeev and Fadeeva [5].

Note that for any matrix A, if all a_{ij} are real, then $A^* = A^T$. Lemma

3.3 will be used in the subsequence development to get the estimate for the improvement in the objective function. Lemmas 3.1 and 3.2 are used to establish the following lemma.

LEMMA 3.4

The inverse of the arithmetically smallest eigenvalue of BB^T is equal to the largest eigenvalue of $(B^{-1})^T(B^{-1})$.

Proof: Let $A = BB^T$, then A is symmetric and $A^{-1} = (B^{-1})^T(B^{-1})$. To show that A is positive definite, observe that

$$X^TAX = X^TBB^TX = (B^TX, B^TX) > 0.$$

Since A is symmetric and positive definite, the lemma follows directly by applying lemmas 3.1 and 3.2.

Having established these lemmas, we now return to the task of finding an estimate for $f(X_B) - f(X_{\overline{B}})$. Going back to equation (4.13), the term $C_BB^{-1}P_s$ may now be estimated as follows:

by lemma 3.3 $\quad |B^{-1}P_s| \leq \sqrt{\lambda_n}\; |P_s|$,

where $\sqrt{\lambda_n}$ is the largest eigenvalue of $(B^{-1})^T(B^{-1})$.

But by lemma 3.5 $\sqrt{\lambda_n} = \dfrac{1}{\sqrt{\lambda_o}}$,

where λ_0 is the smallest eigenvalue of BB^T. Hence,

$$|B^{-1}P_s| \le \frac{|P_s|}{\sqrt{\lambda_0}} \quad ,$$

which gives

$$|C_B^T B^{-1}P_s| \le |C_B|\;|B^{-1}P_s| \le \frac{|C_B|\;|P_s|}{\sqrt{\lambda_0}}$$

Equation (4.13) now reduces to

$$|f(X_B) - f(X_{\bar{B}})| \le \left(\frac{|C_B|\;|P_s|}{\sqrt{\lambda_0}} - C_s\right) \cdot v_s \tag{4.14}$$

Equation (4.14) gives an estimate of the improvement in the objective function between successive iterations in terms of the basis B. A maximal estimate of the right hand side may be obtained by substituting maximum value for $|C_B|$ and a minimum value for λ_0, for all possible bases B. The expression in (4.14) has direct application in linear programming problems that have bounded variables, where a maximal estimate of v_s is readily available. It is also useful in real world situations where a system is presently operating at optimal levels but the cost co-efficients are liable to change. Thus, in a case where some component of C_B increases, or the value C_s for some variable which is not in solution goes down, equation (4.14) may be used directly without solving the entire linear programming problem to determine if the improvement warrants a change in the operating system.

V. UNCONSTRAINED OPTIMIZATION BY GRADIENT METHODS

This chapter will be devoted to the study of the rate of convergence of the gradient method and the general directional method when applied to the quadratic programming problem defined in equation (5.1). The unconstrained quadratic programming problem may be defined as

$$\text{Min} \quad f(X) = (C, X) + \frac{1}{2}(AX, X) \ , \tag{5.1}$$

where C is an n dimensional vector

and A is a symmetric nxn positive definite matrix.

The gradient method provides an iterative procedure to determine the optimal vector X^* which minimizes equation (5.1). Given a point X_k, the gradient method defines the next point X_{k+1} as

$$X_{k+1} = X_k + tv \ , \tag{5.2}$$

where v is a direction vector and t is a scalar.

In the gradient method, the direction v is given by the gradient vector $\nabla f(X_k)$ and the scalar t is taken so as to minimize $f(X_{k+1})$ in the direction v. In literature this method is generally called the "optimal gradient method". There are search techniques that proceed in a direction v which is other than the gradient direction. The scalar t for these techniques is again taken so as to minimize $f(X_{k+1})$ in the direction v. Any algorithm meeting these two conditions will be referred to as the "general directional method". The rate of convergence for the gradient method was first obtained by Kantorovich [10] and is described by Fadeev and Fadeeva [5]. Kantorovich, in trying to find a solution to a linear system

$$BX = F \; , \tag{5.3}$$

where B is a (nxn) positive definite matrix and F is a n-vector, showed that the problem is related to finding a minimum for a function

$$g(X) = (BX,\, X) - 2(F,\, X) \; . \tag{5.4}$$

Equation (5.4) is a quadratic programming problem. The vector X^* which minimizes g(X) is also a solution to equation (5.3). The rates of convergence obtained by the author for the general equation (5.1) are identical to those of Kantorovich and the development differs only slightly from that of Kantorovich. The development presented in this chapter carries an additional term, the net result of which is that equation 5.12 is an inequality for this case, whereas it was an equality for Kantorovich. The expression for the rate of convergence obtained for the gradient method is further generalized for the general directional method.

In the development that follows it is assumed, without loss of generality, that $f(X^*) = 0$ for equation (5.1). The inequality in Lemma 5.1 is due to Kantorovich and will be used to arrive at the expression for the rate of convergence. The proof of the lemma is presented in Fadeev and Fadeeva [5].

LEMMA 5.1 The inequality

$$\frac{\left(\sum_{i=1}^{n} a_i\right)^2}{\left(\sum_{i=1}^{n} \gamma_i a_i\right)\left(\sum_{i=1}^{n} \frac{1}{\gamma_i} \cdot a_i\right)} \geq \frac{4}{\left[\sqrt{\frac{m}{M}} + \sqrt{\frac{M}{m}}\right]^2}$$

holds for $a_i > 0$, $\forall_i$

and $0 < m \leq \gamma_i \leq M$, $\forall_i$

1. Gradient Method

The gradient method for equation (5.1) is given by the following equations:

$$f(X_k) = (C, X_k) + \frac{1}{2}(AX_k, X_k)$$

$$\left.\begin{aligned} \nabla f(X_k) &= C + AX_k \\ H(X_k) &= A \text{ for all } k \\ X_{k+1} &= X_k + tv , \end{aligned}\right\} \qquad (5.5)$$

where t is a scalar defined to minimize $f(X_{k+1})$ in the direction $v = \nabla f(X_k)$.

The objective is to find an upper limit for the ratio $\frac{f(X_{k+1})}{f(X_k)}$. The value of $f(X_{k+1})$ may be estimated by the Taylor series expansion as follows:

$$\begin{aligned} f(X_{k+1}) &= f(X_k + tv) \\ &= f(X_k) + (\nabla f(X_k), v) + \frac{1}{2} t^2 (H(X_k)v, v) \\ &= f(X_k) + t(v, v) + \frac{1}{2} t^2 (Av, v) . \end{aligned} \tag{5.6}$$

The last equation is obtained by using the relations in equation 5.5.

The value of t is chosen to minimize $f(X_{k+1})$.

Hence

$$0 = \frac{\partial f(X_{k+1})}{\partial t} = (v, v) + t(Av, v)$$

giving

$$t = - \frac{(v, v)}{(Av, v)} . \tag{5.7}$$

This value of t gives the minimum value of $f(X_{k+1})$ as

$$f(X_{k+1}) = f(X_k) - \frac{1}{2} \frac{(v, v)^2}{(Av, v)} .$$

The desired ratio may now be written as

$$\frac{f(X_{k+1})}{f(X_k)} = 1 - \frac{1}{2}\frac{(v,\ v)^2}{(Av,\ v)f(X_k)}$$

$$= 1 - \frac{1}{2}\frac{(v,\ v)^2}{(Av,\ v)[(C,\ X_k) + \frac{1}{2}(AX_k,\ X_k)]}\ . \qquad (5.8)$$

Now,

$$(C,\ X_k) + \frac{1}{2}(AX_k,\ X_k) = \frac{1}{2}[(C,\ X_k) + (C,\ X_k) + (AX_k,\ X_k)]$$

$$= \frac{1}{2}[(C,\ X_k) + [(AX_k + C),\ X_k)]$$

$$= \frac{1}{2}[(C,\ X_k) + (v,\ X_k)]\ .$$

Substituting the above equality in equation (5.8) gives

$$\frac{f(X_{k+1})}{f(X_k)} = 1 - \frac{(v,\ v)^2}{(Av,\ v)[(C,\ X_k) + (v,\ X_k)]}\ . \qquad (5.9)$$

To get a numerical estimate of the ratio in the left hand side of equation (5.9), the right hand side will be expressed in terms of the eigenvalues and eigenvectors of the matrix A.

Let $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ be the eigenvalues of the matrix A. Since A is symmetric, the eigenvalues are real and since A is positive definite the eigenvalues are positive, thus

$\lambda_i > 0$ and real for all i.

Define U_1, U_2 ... U_n be the normalized eigenvectors associated with these eigenvalues. These vectors are mutually orthogonal and span the space. Because they are normalized,

$$(U_i, U_i) = 1 \quad \text{for} \quad i = 1, 2, \ldots, n.$$

Since the eigenvectors span the space, the vectors v, X_k and C may be expressed as linear combinations of the eigenvectors. Thus define

$$v = \sum_{i=1}^{n} a_i U_i$$

$$X_k = \sum_{i=1}^{n} b_i U_i$$

and $$C = \sum_{i=1}^{n} c_i U_i \ .$$

Using the above relations we may rewrite the terms in the right hand side of equation (5.9) as follows:

$$Av = \sum_{i=1}^{n} a_i \lambda_i U_i$$

$$(v, v) = \sum_{i=1}^{n} a_i^{2}$$

$$(Av, v) = \sum_{i=1}^{n} \lambda_i a_i^{2}$$

$$(C, X_k) = \sum_{i=1}^{n} b_i c_i$$

$$(X_k, v) = \sum_{i=1}^{n} a_i b_i \quad .$$

Substituting the values above in equation (5.9) results in

$$\frac{f(X_{k+1})}{f(X_k)} = 1 - \frac{\left(\sum_{i=1}^{n} a_i\right)^2}{\left(\sum_{i=1}^{n} \lambda_i a_i^{2}\right)\left(\sum_{i=1}^{n} b_i c_i + \sum_{i=1}^{n} a_i b_i\right)} \quad . \tag{5.10}$$

Since $v = C + AX_k$, it implies that $a_i = c_i + \lambda_i b_i$ or that $b_i = (a_i - c_i)/\lambda_i$.

Substituting the above in equation (5.10) gives

$$\frac{f(X_{k+1})}{f(X_k)} = 1 - \frac{\left(\sum_{i=1}^{n} a_i^2\right)^2}{\left(\sum_{i=1}^{n} \lambda_i a_i^2\right)\left(\sum_{i=1}^{n} \frac{(a_i - c_i)c_i}{\lambda_i} + \sum_{i=1}^{n} a_i \frac{(a_i - c_i)}{\lambda_i}\right)}$$

$$= 1 - \frac{\left(\sum_{i=1}^{n} a_i^2\right)^2}{\left(\sum_{i=1}^{n} \lambda_i a_i^2\right)\left(\sum_{i=1}^{n} \frac{a_i^2}{\lambda_i} - \sum_{i=1}^{n} \frac{c_2^{\,2}}{\lambda_i}\right)} \tag{5.11}$$

$$\leq 1 - \frac{\left(\sum_{i=1}^{n} a_i^2\right)^2}{\left(\sum_{i=1}^{n} \lambda_i a_i^2\right)\left(\sum_{i=1}^{n} \frac{a_i^2}{\lambda_i}\right)} . \tag{5.12}$$

The inequality in (5.12) is obtained by dropping the last term in the denominator of (5.11). Kantorovich, in his development had an equality for (5.12). This is the only significant difference between the two developments.

In (5.12) the ratio of $f(X_{k+1})$ to $f(X_k)$ is in a form which permits the direct application of lemma 5.1. Using lemma 5.1 with $m = \lambda_1$ and $M = \lambda_n$ gives

$$\frac{f(X_{k+1})}{f(X_k)} \leq 1 - \frac{4}{\left[\sqrt{\frac{\lambda_1}{\lambda_n}} + \sqrt{\frac{\lambda_n}{\lambda_1}}\right]^2}$$

$$= 1 - \frac{4}{[\frac{\lambda_1}{\lambda_n} + \frac{\lambda_n}{\lambda_1} + 2]}$$

$$= \frac{[\frac{\lambda_1}{\lambda_n} + \frac{\lambda_n}{\lambda_1} - 2]}{[\frac{\lambda_1}{\lambda_n} + \frac{\lambda_n}{\lambda_1} + 2]} = [\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}]^2 .$$

Thus,

$$f(X_{k+1}) \leq [\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}]^2 f(X_k)$$

or

$$f(X_{k+1}) \leq [\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}]^{2(k+1)} f(X_0) \qquad (5.13)$$

Inequality (5.13) gives the rate of convergence for the gradient method. The expression in (5.13) obtained for equation (5.1) is identical to that obtained by Kantorovich for equations (5.3) and (5.4). The rate of convergence obtained above may be expressed in the following theorem:

Theorem 5.1 The gradient method converges to the optimal of the unconstrained quadratic programming problem min $f(X) = (C, X) + \frac{1}{2}(AX, X)$ with a rate of geometric progression. The rate of convergence is measured by $[\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}]^2$, where λ_1 and λ_n are the minimum and maximum eigenvalues of the matrix A.

A similar development may now be used for the general directional method.

2. General Directional Method

The general directional method differs from the gradient method in that the direction vector v need not necessarily be the gradient direction. Equation (5.5) holds for this method except that in general $v \neq \nabla f(X_k)$. The development in this section is very similar to the development in the last section. The procedure once again is to get the Taylor series expansion for $f(X_{k+1})$ and use this to find an upper bound for the ratio of $f(X_{k+1})$ to $f(X_k)$. The Taylor series expansion for $f(X_{k+1})$ is given by

$$\begin{aligned} f(X_{k+1}) &= f(X_k + tv) \\ &= f(X_k) + t(\nabla f(X_k), v) + \frac{1}{2} t^2(H(X_k)v, v) \\ &= f(X_k) + t[(C + AX_k), v] + \frac{1}{2} t^2(Av, v) . \end{aligned}$$

As in the previous section, the value of t is determined so as to minimize $f(X_{k+1})$.

Thus

$$0 = \frac{\partial f(X_{k+1})}{\partial t} = [(C + AX_k), v] + t(Av, v) ,$$

or

$$t = - \frac{[(C + AX_k), v]}{(Av, v)} . \tag{5.14}$$

For the above value of t, the value of $f(X_{k+1})$ is given by

$$f(X_{k+1}) = f(X_k) - \frac{[(C + AX_k), v]^2}{2(Av, v)} .$$

or

$$\frac{f(X_{k+1})}{f(X_k)} = 1 - \frac{[(C + AX_k), v]^2}{2(Av, v)f(X_k)}$$

$$= 1 - \frac{[(\nabla f(X_k), v)]^2}{(Av, v)[(C, X_k) + (\nabla f(X_k), X_k)]} \tag{5.15}$$

In order to express (5.15) in the form of equation (5.9) in the previous section, we represent the directional vector v as

$$v = \nabla f(X_k) \text{ Cos } \Theta_k ,$$

where Θ_k is the angle between the direction v and the gradient at X_k. Equation (5.15) may now be rewritten as:

$$\frac{f(X_{k+1})}{f(X_k)} = 1 - \frac{(\nabla f(X_k), \nabla f(X_k))^2 \operatorname{Cos}^2\Theta_k}{(Av, v)[(C, X_k) + (\nabla f(X_k), X_k)]} \tag{5.16}$$

Equations (5.9) and (5.16) differ only by the $\operatorname{Cos}^2\Theta_k$ term in the numerator of (5.16). In view of this, the development of the previous section may be used to obtain the rate of convergence for this method. Accordingly, define Q as the last term in equation (5.9), then (5.16) may be rewritten as

$$\frac{f(X_{k+1})}{f(X_k)} = 1 - Q \operatorname{Cos}^2\Theta_k \ . \tag{5.17}$$

We now need an estimate of $(1 - Q \operatorname{Cos}^2\Theta_k)$. This estimate may be obtained from equations (5.9) and (5.13) of the previous section. We have

$$1 - Q \leq [\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}]^2$$

or

$$1 - Q \operatorname{Cos}^2\Theta_k \leq 1 - \operatorname{Cos}^2\Theta_k + [\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}]^2 \operatorname{Cos}^2\Theta_k \ . \tag{5.18}$$

For notational convenience define

$$r_k = 1 - \operatorname{Cos}^2\Theta_k + [\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}]^2 \operatorname{Cos}^2\Theta_k$$

and $\bar{r} = \max_k \{r_k\}$.

Then using equations (5.17), (5.18) and the above definitions gives

$$\frac{f(X_{k+1})}{f(X_k)} \leq r_k$$

or

$$f(X_{k+1}) \leq (\bar{r})^{k+1} f(X_0) \ . \qquad (5.19)$$

Equation (5.19) indicates that the process will converge at the rate of geometric progression as long as $\bar{r} < 1$. These results are summarized in the following theorem.

Theorem 5.2

A directional procedure moving in the direction converges to the optimal of the quadratic function defined by (5.1) at the rate of geometric progression. The rate of convergence may be measured by

$$1 - \text{Cos}^2\,\bar{\Theta} + [\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}]^2\ \text{Cos}^2\bar{\Theta}$$

where $\bar{\Theta}$ is the maximum of the angles Θ_k between the gradient and the direction v and, λ_1 and λ_n are the minimum and maximum eigenvalues of the matrix A.

VI. ROSEN'S METHOD

The rate of convergence of the gradient method used for the solution of the unconstrained quadratic programming problem was discussed in Chapter V. A procedure very close to the gradient method will now be discussed for the solution of quadratic programming problem with constraints. Rosen's gradient projection method [15], [12], consists of projecting the gradient onto the boundary of the feasible domain and then proceeding along the projection. If the gradient lies within the feasible domain, then no projection is necessary. Hence in the interior of the feasible domain, Rosen's method coincides with the gradient method.

More specifically, if X_k be the solution vector after the k^{th} iteration, then Rosen's method consists of the following steps:

1. Calculate the gradient vector and check for optimality. If the solution is optimal, stop. If not, proceed to step 2.

2. If the gradient vector points inside the feasible domain, proceed to step 3. Otherwise, project the gradient on a manifold of least dimension containing X_k.

3. Proceed along the direction obtained in step 2 so as to decrease the objective function (if this is not possible for the projected gradient, appropriately increase the dimension of the manifold containing X_k by 1 and return to step 2) until a minimum along the direction is reached or a constraint boundary is encountered. Call this point X_{k+1} and return to step 1.

For the purposes of this development, a verbal description of Rosen's method is sufficient; a detailed algebraic description may be found in Rosen's original article [15]. Rosen showed in his paper that under the following assumptions:

a) the starting point is in the interior of the feasible region;

b) the iterative procedure never requires projection;

c) the objective function is quadratic;

d) optimal gradient step is taken between iterations;

e) the optimal value of the objective function is zero.

The rate of convergence is estimated by:

$$0 \leq \frac{f(X_{k+1})}{f(X_k)} \leq 1 - \frac{\lambda_1^2}{\lambda_n^2} \qquad (6.1)$$

where λ_1 and λ_n are the minimum and maximum eigenvalues of the symmetric positive definite matrix A defined in equation (5.1).

It is clear that under the above assumptions the procedure reduces to the optimal gradient method described in Chapter V. Hence theorem 5.1 may be directly applied to obtain the rate of convergence under the assumptions given above.

Since

$$1 - \frac{\lambda_1^2}{\lambda_n^2} \geq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \qquad (6.2)$$

Theorem 5.1 gives a tighter bound on the rate of convergence as follows:

$$0 \leq \frac{f(X_{k+1})}{f(X_k)} \leq (\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1})^2 \tag{6.3}$$

However, both (6.1) and (6.3) are limited to the case where no projections are permitted. In order to generalize the above equations we need to establish the following theorems.

Theorem 6.1 If λ_1 and λ_n are the minimum and maximum eigenvalues of operator A, then

$$\lambda_1 = \min_{X \neq 0} \frac{(AX, X)}{(X, X)}$$

$$\lambda_n = \max_{X \neq 0} \frac{(AX, X)}{(X, X)}$$

Proof: Follows from theorems 11.6 and 11.7 in Fadeev and Fadeeva [5].

Theorem 6.2 Given a (nxn) symmetric positive definite matrix A of full rank and a linear transformation $\Phi(X)$ from E^n to E^m of rank $m < n$, then $\Phi(X) = 0$ defines a manifold Q in which the following inequality holds

$$(\frac{c_n - c_1}{c_n + c_1})^2 \_ (\frac{b_n - b_1}{b_n + b_1})^2 \tag{6.4}$$

where,

$$b_1 = \min_{X\neq 0} \frac{(AX, X)}{(X, X)}, \quad X\varepsilon E^n$$

$$b_n = \max_{X\neq 0} \frac{(AX, X)}{(X, X)}, \quad X\varepsilon E^n$$

$$c_1 = \min_{X\neq 0} \frac{(AX, X)}{(X, X)}, \quad X\varepsilon Q$$

$$c_n = \max_{X\neq 0} \frac{(AX, X)}{(X, X)}, \quad X\varepsilon Q$$

Proof: By definition, $b_1 \leq b_n$ and $c_1 \leq c_n$. Since Q is a proper subset of E^n it follows that

$$\min_{X\neq 0} \frac{(AX, X)}{(X, X)}, \quad X\varepsilon E^n \quad \leq \quad \min_{X\neq 0} \frac{(AX, X)}{(X, X)}, \quad X\varepsilon Q$$

and

$$\max_{X\neq 0} \frac{(AX, X)}{(X, X)}, \quad X\varepsilon E^n \quad \geq \quad \max_{X\neq 0} \frac{(AX, X)}{(X, X)}, \quad X\varepsilon Q$$

hence,

$$b_1 \leq c_1 \leq c_n \leq b_n \tag{6.5}$$

Since A is positive definite b_1, c_1, b_n, c_n are all positive, so that the squares in the inequality 6.4 may be dropped. It is now only necessary to show that

$$\frac{c_n - c_1}{c_n + c_1} \leq \frac{b_n - b_1}{b_n + b_1} \tag{6.6}$$

or that $(c_n - c_1)(b_n + b_1) \leq (c_n + c_1)(b_n - b_1)$

or that $b_1 c_n \leq c_1 b_n$. (6.7)

Inequality (6.7) follows directly from (6.5), thereby establishing the theorem.

In view of the fact that the projected gradient is the direction of steepest descent within the manifold into which the projection occurs, applying theorem 5.1 gives

$$0 \leq \frac{f(X_{k+1})}{f(X_k)} \leq \left(\frac{c_n - c_1}{c_n + c_1}\right)^2 . \qquad (6.8)$$

Also by theorem 6.2 we have

$$\left(\frac{c_n - c_1}{c_n + c_1}\right)^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 . \qquad (6.9)$$

It follows from equations (6.8) and (6.9) that

$$0 \leq \frac{f(X_{k+1})}{f(X_k)} \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 . \qquad (6.10)$$

Equation 6.10 shows that both in the case of unprojected gradient search and for the projected gradient search within a given manifold, the rate of convergence defined by (6.3) is applicable. Thus (6.3) gives the rate of convergence for all iterations of Rosen's method except those that call for a change in manifold on which the projection occurs. More generally, the upper and lower bounds for convergence of the unconstrained directional search are applicable to all procedures that restrict the search to

a) Directional search within the feasible region never encountering the constraint boundary except possibly at the optimal

b) Directional search within any single constraint boundary.

VII. CONCLUSION

1. Summary

The rate of convergence is a useful measure of the performance of an algorithm. Knowledge of the rate can help determine which algorithm is best suited for a given problem. This research is a study of the rate of convergence of a few algorithms used for nonlinear programming problems. The Newton-Raphson procedure and a higher order procedure used for the solution of nonlinear equations is studied. Both the convergence and the rate of convergence for the multivariate Newton-Raphson procedure is presented in the simple format of the Newton-Raphson procedure for scalar functions. A higher order procedure, which results directly from Taylor series expansion is presented. Its convergence is established and a measure for the rate of convergence is obtained. A multivariate generalization of this higher order procedure is seen to have little practical value.

In analyzing the simplex algorithm, it was not possible to obtain an expression for its rate of convergence, however, an expression for the improvement in the objective function between successive iterations is obtained. This expression is entirely in terms of the original problem rather than intermediate computations.

The bound, due to Kantorovich, for the rate of convergence of the optimal gradient method used in solution for a system of linear equations is shown to hold for the general unconstrained quadratic programming problem. The result is then extended for the general directional procedure.

Rosen presented a bound for the rate of convergence of his algorithm. The bound was obtained under very strict assumptions of the computational procedure. It is seen that under the same assumptions, tighter bounds are available for Rosen's method and that these bounds are also applicable under less stringent assumptions about the computational procedure.

2. Suggestions for Further Study

There are numerous algorithms in nonlinear programming for which the rate of convergence has not been established. Algorithms by Beale, Barkin and Dorfman, Frisch and by Zoutendijk serve as examples. A study of the rate of convergence of these algorithms will be extremely helpful to users in choosing an algorithm for their specific problems.

Along the lines of this research, a promising area for further study may be defined in terms of the following new concepts.

Assuming a constrained optimization problem where the unconstrained optimal X^* lies outside the convex feasible region S, then a point V of the feasible region will be called "visible" if the line joining X^* to V lies entirely outside S.

Thus the point $V \varepsilon S$ is visible if the intersection of the set S and the set L is a null set, where

$$L\{X | X = X^* + t(V - X^*),\ 0 \leq t < 1\} .$$

Let Q be a set of all visible points in S. Let $H : CX = 0$ be a supporting hyperplane to the set Q such that the half space $H_1 : CX \geq 0$ containing Q also contains X^*, then the intersection $H_1 \cap S$ will be called the visible region.

With the above definitions possibly one can establish the following propositions for Rosen's method applied to quadratic programming problem.

1. Proposition 1: There exists an index k such that for all $i > k$, X_i lies on the boundary of S.
2. Proposition 2: The index k is the smallest i for which X_i is included in the visible region.

   If Proposition 2 can be shown to be true, then proposition 3 may be stated as follows.
3. Proposition 3: There exists a 1-1 mapping between the set Q and the supporting hyperplane H contained in the visible region.

Proposition 1 is trivially true for the boundary containing the constrained optimal. If propositions 2 and 3 can be established, then it may be possible to restrict the search of Rosen's method to the supporting hyperplane. If this can be done, it will permit a convenient analysis for the rate of convergence of Rosen's algorithm.

REFERENCES

1. H. Akaike, 1959, On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method, Ann. Inst. Stat. Math., Tokyo 11, 1-16.

2. I. S. Berezin and N. P. Zhidkov, 1965, Computing methods, Vol. II (Addison-Wesley) Ch. 6, 7.

3. M. D. Canon and C. Cullum, 1969, Upper and lower bounds on the rate of convergence of the Frank-Wolfe algorithm, SIAM J. Control.

4. G. B. Dantzig, 1963, Linear programming and extensions (Princeton Univ. Press).

5. D. K. Fadeev and V. N. Fadeeva, 1963, Computational methods of linear algebra (W. H. Freeman).

6. W. H. Fleming, 1965, Functions of several variables (Addison-Wesley).

7. J. L. Greenstadt, 1966, A ricocheting gradient method for non-linear optimization, SIAM Appl. Math., Vol. 14, No. 3.

8. G. Hadley, 1961, Linear Algebra (Addison-Wesley).

9. F. John, 1967, Lectures on advanced numerical analysis (Gordon and Breach) Ch. 2.

10. L. W. Kantorovich and G. P. Akilov, 1964, Functional analysis in normed spaces (Macmillan, New York) Ch. 15.

11. V. Klee, 1966, Pivot Rules in Linear Programming, Presented at twenty-ninth National ORSA meeting, Santa Monica, California.

12. H. P. Kunzi and W. Krelle, 1966, Nonlinear Programming (Blaisdell Publishing).

13. N. Macon, 1963, Numerical Analysis (John Wiley).

14. O. L. Magasarian, 1969, Nonlinear programming (McGraw Hill).

15. J. B. Rosen, 1960, The gradient projection method for nonlinear programming. Part I. Linear constraints, J. Soc. Indust. Appl. Math. Vol. 8, No. 1.

16. D. J. Wilde and C. S. Beightler, 1967, Foundations of Optimization (Princeton-Hall).

17. P. Wolfe, 1970, Convergence theory in nonlinear programming in: Integer and Nonlinear programming, ed. J. Abadie, (North-Holland, Amsterdam) Ch. 1.

18. W. I. Zangwill, 1969, Nonlinear Programming: A unified approach (Prentice-Hall).

VITA

Vinod Chachra, son of Mr. and Mrs. N. L. Chachra, was born on August 27, 1945, in Calcutta, India. He graduated from St. Xavier's School in Calcutta in 1962.

He enrolled at The Indian Institute of Technology, Kharagpur, in July 1962 and worked for a Bachelor of Technology degree in Mechanical Engineering which he received in June 1967. He came to the United States in September 1967 and entered Graduate School at Virginia Polytechnic Institute and State University. He completed his requirements for his Master of Science degree in Industrial Engineering in August 1968.

Starting August 1968, he worked in industry with the FMC Corporation at Fredericksburg, Virginia and Philadelphia, Pennsylvania. In September 1969, he entered the Graduate School at Stanford University in Pala Alto, California and worked toward a Doctor of Philosophy degree in Operations Research. He transferred to Virginia Polytechnic Institute and State University in April 1970, and is a candidate for completion of the Doctor of Philosophy in Industrial Engineering and Operations Research degree requirements in June 1972.

The author is a recipient of the B. C. Roy Gold Medal awarded at the Indian Institute of Technology. In the United States he is affiliated with Phi Khappa Phi, Alpha Phi Mu and the American Institute of Industrial Engineers.

V. Chachra

RATE OF CONVERGENCE IN NONLINEAR PROGRAMMING

by

Vinod Chachra

(ABSTRACT)

The rate of convergence is a useful measure of the performance of an algorithm. Knowledge of the rate can help determine which algorithm is best suited for a given problem. This research is a study of the rate of convergence of a few algorithms used for nonlinear programming problems. The Newton-Raphson procedure and a higher order procedure used for the solution of nonlinear equations is studied. Both the convergence and the rate of convergence for the multivariate Newton-Raphson procedure is presented in the simple format of the Newton-Raphson procedure for scalar functions. A higher order procedure, which results directly from Taylor series expansion is presented. Its convergence is established and a measure for the rate of convergence is obtained. A multivariate generalization of this higher order procedure is seen to have little practical value.

In analyzing the simplex algorithm, it was not possible to obtain an expression for its rate of convergence, however, an expression for the improvement in the objective function between successive iterations is obtained. This expression is entirely in terms of the original problem rather than intermediate computations.

The bound, due to Kantorovich, for the rate of convergence of the optimal gradient method used in solution for a system of linear

equations is shown to hold for the general unconstrained quadratic programming problem. The result is then extended for the general directional procedure.

Rosen presented a bound for the rate of convergence of his algorithm. The bound was obtained under very strict assumptions of the computational procedure. It is seen that under the same assumptions, tighter bounds are available for Rosen's method and that these bounds are also applicable under less stringent assumptions about the computational procedure.