UNCONSTRAINED NONLINEAR MINIMIZATION AND THE RICHARDS FUNCTION

by

KANG-HOH PHUA, B.Sc.

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in the Department

of

Mathematics

We accept this thesis as conformin
to the required standard

[redacted]

UNIVERSITY OF VICTORIA

APRIL 1971

ABSTRACT

A survey of unconstrained nonlinear optimization techniques is made. Some of these methods are then applied to solve the problem of fitting the Richards function to experimental data by the method of least-squares.

In attempting to solve this problem, the dimensions of the problem are first reduced from 4 to 2, by solving the linear parameters in terms of the nonlinear parameters from the so-called normal equations. Numerical results show that this approach is more efficient. Moreover, a comparison is made between the performances of the selected algorithms in solving this problem:

Examiners: [redacted]

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to Professor I. Barrodale, my supervisor, for his advice and guidance and for his limitless patience, which have been essential to me in the preparation of this thesis.

Special thanks are also extended to those members of the Department of Mathematics who were so generous with their time and assistance. I also wish to thank Dr. C. Chan for his comments and suggestions, and the staff of the University of Victoria Computer Centre for their help.

Finally, I wish to especially thank Miss B. Cooknell for all her help and for her careful typing of this thesis.

CHAPTER ONE

PRELIMINARIES

1.1. Maxima and Minima.

1.2. Concave and Convex Functions.

1.3. Constrained and Unconstrained Problems.

1.4. Some Difficulties of the Analytical Approach.

1.5. Numerical Optimization Techniques.

1.1 Maxima and Minima.

Let E^n be the usual n-dimensional Euclidean space. An element of E^n will be written as

$$x = (x_1, x_2, \ldots, x_n)$$

throughout this thesis. Consider a real-valued function $f(x)$ defined in E^n. Denote by $D(f)$ the domain of f. A point $x^* \in D(f)$ is said to be a local minimum of the function $f(x)$ if there is a neighborhood $N \subset D(f)$ containing x^* in its interior such that

$$f(x) \geq f(x^*) , \quad \forall x \in N . \tag{1.1}$$

x^* is called a local maximum of the function $f(x)$ if it is a local minimum of $(-f(x))$.

A point $x^* \in D(f)$ is said to be a global minimum of the function $f(x)$ if the inequality (1.1) holds for $N = D(f)$.

1.2. Concave and Convex Functions.

A set S is said to be convex if for any two points, x, y in S and for any $0 \leq \lambda \leq 1$, the point $z = \lambda x + (1-\lambda)y$ is also a member of S.

A function $f(x)$ is said to be convex on a convex region $R \subset D(f)$ if for any two points $x, y \in R$ and any $0 \leq \lambda \leq 1$, the following inequality holds:

$$f[\lambda x + (1-\lambda)y] \leq \lambda f(x) + (1-\lambda)f(y) . \tag{1.2}$$

If $R = D(f)$, then we simply say that $f(x)$ is convex.

The function $f(x)$ is concave if $(-f(x))$ is convex.

The function $f(x)$ is strictly convex if strict inequality holds in (1.2) for any two points x,y in $D(f)$.

In the calculus, we define a point $x \in D(f)$ which satisfies the conditions:

$$\frac{\partial f}{\partial x_i} = 0 , \qquad i = 1, 2, \ldots, n. \tag{1.3}$$

as a stationary point of $f(x)$.

Suppose that x^* is a stationary point of $f(x)$. Then clearly x^* is a local minimum of $f(x)$ if $f(x)$ is convex on a neighborhood of x^*. Similarly, a sufficient condition for x^* to be a local maximum is that $f(x)$ is concave on a neighborhood of x^*. However, if the function $f(x)$ is known to be convex or concave, then x^* must be a global minimum or a global maximum, respectively.

Assume that the second partial derivatives of the function $f(x)$ exist everywhere in $D(f)$. The $n \times n$ Hessian matrix G of the function $f(x)$ at a point $x \in D(f)$ is defined by

$$G = \left[\frac{\partial^2 f}{\partial x_i \, \partial x_j}\right],$$

where all the partial derivatives are evaluated at the point x. It is well-known that $f(x)$ is convex if and only if its Hessian matrix is positive semi-definite at each point in its domain.

1.3. Constrained and Unconstrained Problems.

The problem of minimizing or maximizing a function $f(x)$ for which each value of the components of x can be any real number is called an unconstrained optimization problem. Notice that maximizing a function

$f(x)$ is equivalent to minimizing the function $(-f(x))$; only minimization problems will be considered from here on.

Suppose all the first partial derivatives of $f(x)$ exist. Then the problem of finding a stationary value of a function is equivalent to solving the system of equations defined in (1.3). If the system (1.3) happens to be a set of nonlinear equations, the problem is said to be nonlinear; otherwise it is linear. The problem of solving the system (1.3) can be posed as an optimization problem, since a solution to equations (1.3) also minimizes the function

$$s(x) = \sum_{i=1}^{n} \left\{\frac{\partial f}{\partial x_i}\right\}^2 . \tag{1.4}$$

Consider the classical constrained problem:

minimize $g(x)$

subject to the equality constraints

$$g_j(x) = 0, \quad j = 1,2,\ldots,m.$$

Geometrically, we are looking for a minimum of the function $g(x)$ along the surface (the degenerate cases of curve and point are considered also) intersected by the surfaces $g_j(x) = 0$, $j = 1,2,\ldots,m$. By using the Lagrange multiplier technique, this problem is equivalent to finding a minimum of the unconstrained function:

$$h(x,\lambda) = g(x) + \sum_{j=1}^{m} \lambda_j \, g_j(x) \tag{1.5}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ is a set of Lagrange multipliers. Thus the classical constrained problem can be approached either

(i) by solving the system of equations

$$\frac{\partial g}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \frac{\partial g_j}{\partial x_i} = 0 , \qquad i = 1,2,\ldots,n$$

$$g_j = 0 , \qquad j = 1,2,\ldots,m$$

or

(ii) by minimizing the unconstrained function

$$f(x,\lambda) = \sum_{i=1}^{n} \left\{ \frac{\partial g}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \frac{\partial g_j}{\partial x_i} \right\}^2 + \sum_{j=1}^{m} g_j^{\,2} . \qquad (1.6)$$

The general constrained optimization problem takes the form

minimize $g(x)$

subject to the constraints

$$g_j(x) \leq 0 , \; j = 1,2,\ldots,m$$

and

$$x_i \geq 0 , \; i = 1,2,\ldots,n .$$

Note that the nonnegativity constraints are not restrictive, since if (say) x_1 is not nonnegative it can be replaced by $x_1 = y_1 - z_1$, where $y_1 \geq 0$, $z_1 \geq 0$. This is also known as the mathematical programming problem. A variety of numerical techniques are available for solving special cases of this problem: linear programming, quadratic programming, etc. Discussion in detail of this problem is not within the scope of this thesis.

1.4. Some Difficulties of the Analytical Approach.

A minimum of a function $f(x)$ with continuous first partial derivatives in its domain $D(f)$ is characterized by the classical theory of unconstrained optimization as follows.

A necessary condition is that it satisfies the equations (1.3). A sufficient condition for a point satisfying (1.3) to be a minimum is that all the second partial derivatives exist at this point and the Hessian matrix at it is positive semi-definite.

From a practical computational view-point, the difficulties associated with this approach are several.

First, the system (1.3) is usually a set of n nonlinear equations in the n unknowns x_i. Unfortunately the task of solving large sets of nonlinear equation is very difficult. The function $f(x)$ may be so complex that it is difficult even to write out (1.3) in closed form.

Secondly, even when we can solve the system (1.3), there is no guarantee that a given solution is not a maximum, a saddle point, etc., rather than a global minimum. These situations are illustrated for a single variable function $f(x)$ in Fig. 1.1.

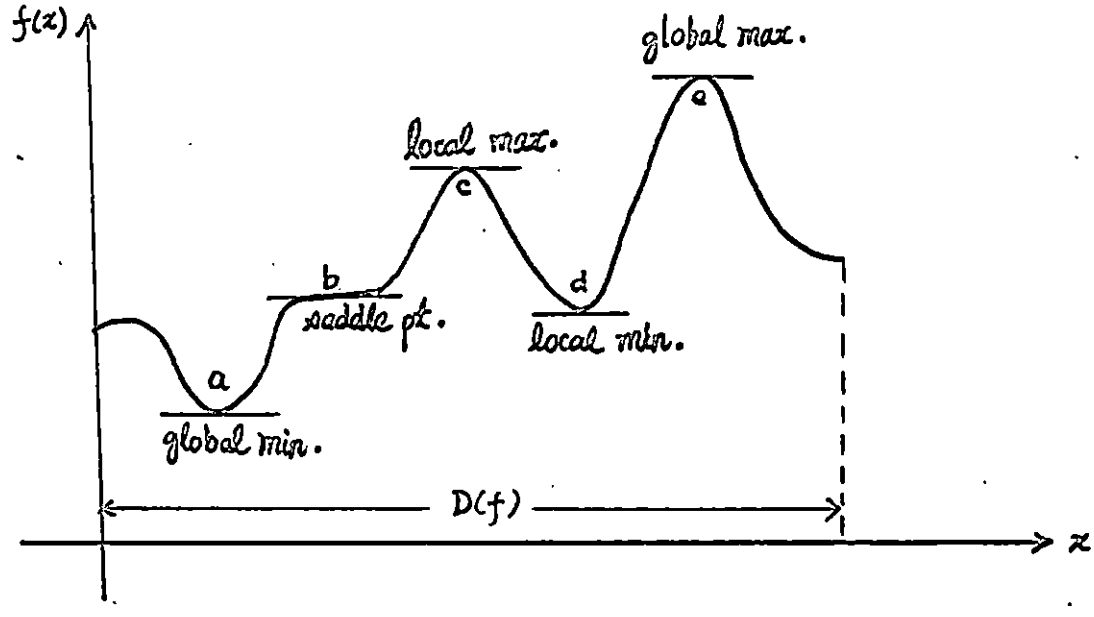

Figure 1.1. A function having several types of stationary points.

Here a, b, c, d and e are points at which the equations (1.3) are satisfied, while only the point a is actually desired.

Thirdly, the total number of stationary points of the function $f(x)$ cannot usually be determined from these equations. Furthermore, the number of stationary points may be so large (sometimes infinite) that it is impractical to attempt to identify a global minimum or maximum.

Finally, we note that the analytical approach is not readily applicable to functions with discontinuous derivatives as, for example, the function

$$f(x) = \sum_{i=1}^{N} |f_i(x)|.$$

Such functions might have well-defined optima, and they often occur in practice.

Thus other approaches must be considered.

1.5. Numerical Optimization Techniques.

Most of the numerical optimization techniques have certain features in common. For example, they all require an initial point $x^{(0)}$ to be specified, and then they proceed by generating a sequence of points $x^{(i)}$, $i = 1,2,\ldots$ such that

$$f(x^{(i+1)}) \leq f(x^{(i)}) , \tag{1.7}$$

These techniques are referred to as iterative techniques, and they can be studied with the aid of the equation

$$x^{(i+1)} = x^{(i)} + \lambda_i d_i , \tag{1.8}$$

where d_i is an n-dimensional direction vector and λ_i is a distance moved along this direction.

Iterative techniques can be subdivided into two classes:

Direct search methods are those which do not require the explicit evaluation of any partial derivatives of the function which is to be minimized. They make use of the function values plus the information gained from earlier iterations in determining the successive values for $x^{(i)}$. In fact, some of these methods use these function values to obtain

numerical approximations of the derivatives of the function, or, to fit low degree polynomials or surfaces through selected points.

Gradient methods are those methods in which values of the partial derivatives of the function to be minimized are required. These values together with the function value itself, plus information obtained from earlier iterations, are used to create the direction d_i and λ_i in equation (1.7).

Numerical optimization techniques are important not only because they can be applied to solve many difficult practical problems but also because they provide the numerical analyst with a tool for tackling a wide range of nonlinear problems.

CHAPTER TWO

DIRECT SEARCH METHODS

2.1. Introduction.

As defined in section 1.5, direct search methods are those minimization techniques not requiring explicit evaluation of any derivatives of the objective function. Methods of this type are useful in the early stages of optimization, and they can provide information about a region in which a minimum is located. In general, they do not give a rapid rate of convergence and hence are not efficient for finding a minimum with high precision.

There are many methods in this class, for example, random search, Fibonacci search, simplex method, etc. The one we use in this thesis was suggested by Powell [8], and it is probably the most effective direct search method presently available [4].

We shall begin the discussion of this method by considering the simple case of minimizing a function of one variable. As we shall see, this problem is not completely divorced from the general problem, since many of the techniques for minimizing a multi-variate function perform a sequence of linear searches along various directions. Each of these linear searches is equivalent to a univariate search.

2.2. Univariate Search.

Let $f(x)$ be a function defined in E^1 and x_1, x_2, x_3 any three distinct points in E^1 with corresponding function values f_1, f_2, f_3 respectively. Consider the quadratic function

$$q(x) = ax^2 + bx + c \tag{2.1}$$

which passes through these three points. Define

$$a_1 = \begin{vmatrix} f_1 & x_1 & 1 \\ f_2 & x_2 & 1 \\ f_3 & x_3 & 1 \end{vmatrix} \quad , \quad b_1 = \begin{vmatrix} x_1^2 & f_1 & 1 \\ x_2^2 & f_2 & 1 \\ x_3^2 & f_3 & 1 \end{vmatrix}$$

$$c_1 = \begin{vmatrix} x_1^2 & x_1 & f_1 \\ x_2^2 & x_2 & f_2 \\ x_3^2 & x_3 & f_3 \end{vmatrix} \quad , \quad d = \begin{vmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{vmatrix}$$

Then a, b, c in equation (2.1) are given by:

$$a = a_1/d \ , \ b = b_1/d \ , \ c = c_1/d$$

The stationary point of the quadratic is:

$$x_m = -\frac{b}{2a} = \frac{1}{2} \frac{(x_2^2 - x_3^2)f_1 + (x_3^2 - x_1^2)f_2 + (x_1^2 - x_2^2)f_3}{(x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3} \qquad (2.2)$$

Now x_m is a minimum of $q(x)$ if $a > 0$; otherwise it is a maximum or a saddle point.

Suppose that $f(x)$ is the function we want to minimize. Let x_1 be an initial point approximating the minimum, h being the step-size, and ϵ the required accuracy in determining the optimal point. In Powell's algorithm [8] for univariate search, the function $f(x)$ is first evaluated at x_1 and at the point $x_2 = x_1 + h$, to give f_1 and f_2, respectively. Define

$$x_3 = \begin{cases} x_1 + 2h & \text{if } f_1 \geq f_2 \\ x - h & \text{if } f_1 < f_2 \end{cases}$$

Evaluate the function at x_3 to give f_3. Then a quadratic of the form (2.1) is fitted through these three points. The predicted minimum, x_m,

can be found by (2.2). Determine the point corresponding to the smallest values in $\{f_1, f_2, f_3\}$, say x_1. If $|x_m - x_1| < \varepsilon$, the minimum is assumed to be located. Otherwise the function is evaluated at x_m to give f_m. One of the three points x_1, x_2, x_3 is discarded as it corresponds to the greatest function value. This process of quadratic interpolation is then repeated again and again.

An additional remark is relevant here. The maximum step length which is allowed along the direction of search must be provided by the user. If the value of x_m predicted by (2.2) is greater than the maximum step length allowed, or x_m turns out not to be a minimum, i.e.

$$\frac{(x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3}{(x_1 - x_2)\ (x_2 - x_3)\ (x_3 - x_1)} \geq 0$$

then the maximum step length allowed is taken along the direction of decreasing f, and the point so obtained is used to replace one of x_1, x_2 and x_3 as before. Notice that the function value which is thrown away out of $\{f_1, f_2, f_3\}$ is usually the greatest, but this is not the case if rejecting a smaller one enables us to "bracket" a minimum. The main difficulty here is in recognizing when this situation has arisen, and in deciding which function value should be discarded in such a situation.

Refinement of this algorithm will be discussed in section 2.4.

2.3. Powell's Direct Search Algorithm.

Consider the quadratic function in E^n defined by

$$f(x) = xAx + bx + c$$

Two nonzero vectors u and v in E^n are said to be mutually conjugate

if $uAv = 0$. In order that $f(x)$ has a minimum, it is necessary that A be positive definite.

The method due to Powell [8] is based on the procedure for setting up mutually conjugate directions, and each iteration starts with a search down these directions. Powell has shown that if $d_1, d_2, \ldots, d_n$ are n mutually conjugate directions in E^n, then the minimum of the quadratic $f(x)$ may be found by searching along each of the directions once only. Furthermore, if x_0 and x_1 are both minima of $f(x)$ in a space containing the direction d, then the direction $(x_1 - x_0)$ and d are mutually conjugate.

The algorithm starts with an arbitrary initial approximation to the minimum, say x_0, and the directions $d_1^1, d_2^1, \ldots, d_n^1$ are chosen initially to be the coordinate directions in E^n. An iteration of the basic procedure during the k^{th} stage is as follows:

(i) For $r = 1,2,\ldots,n$, use a univariate search technique to calculate λ_r such that $f(x_{r-1} + \lambda_r d_r^k)$ is a minimum, and define $x_r = x_{r-1} + \lambda_r d_r^k$.

(ii) Find the integer m, $1 \leq m \leq n$, such that
$$\Delta = f(x_{m-1}) - f(x_m) = \max_{1 \leq i \leq n} f(x_{i-1}) - f(x_i) .$$

(iii) Evaluate $f_3 = f(2x_n - x_0)$ and define $f_1 = f(x_0)$, $f_2 = f(x_n)$.

(iv) If $f_3 \geq f_1$ or $(f_1 - 2f_2 + f_3) \cdot (f_1 - f_2 - \Delta)^2 \geq \frac{1}{2}\Delta\,(f_1 - f_3)^2$ then set
$$(d_1^{k+1}, d_2^{k+1}, \ldots, d_n^{k+1}) = (d_1^k, d_2^k, \ldots, d_n^k)$$
and $x_0 = x_n$.

(v) Otherwise, define $d = (x_n - x_0)/\|x_n - x_0\|$ and calculate λ such that $f(x_n + \lambda d)$ is a minimum, and set

$$(d_1^{k+1}, d_2^{k+1}, \ldots, d_n^{k+1}) = (d_1^k, d_2^k, \ldots, d_{m-1}^k, d_{m+1}^k, \ldots, d_n^k, d)$$

and $x_0 = x_n + \lambda d$.

(vi) Convergence is assumed to have occurred when an iteration changes each variable by less than the required accuracy.

However, the following ultimate convergence criterion is recommended by Powell:

(a) Continue with the above procedure until an iteration changes each variable by less than one-tenth of the required accuracy. Denote this first solution by a.

(b) Increase each variable by ten times the required accuracy. Restart the procedure until the same situation as (a) has been reached. Denote this second solution by b.

(c) The function is then minimized along the direction (a - b) to give the point c.

(d) If all the components of (a - c) or all the components of (b - c) are less than one-tenth of the required accuracy, ultimate convergence is assumed.

(e) Otherwise, include the direction (a - c) into the set of search directions in place of the current d_1, and restart the whole process from the point c by returning to (a).

This is a very safe criterion and it has proved to be satisfactory in practice. However, it has been found that the first solution a is usually sufficiently close to the minimum for most practical purposes. A considerable reduction in function evaluations is achieved if the searches for b and c are omitted.

2.4. Some Additional Remarks on Powell's Direct Search Algorithm.

Powell's method has been shown to be inefficient on some problems where it fails to choose any new directions and so reduces to an alternating variable search [4].

Zangwill [17] has made some criticisms of Powell's method; in particular he considers the function $f(x, y, z) = (x - y + z)^2 + (-x + y + z)^2 + (x + y - z)^2$, where x, y and z are independent variables. Here, f is a strictly convex function with a unique minimum at the point $(0, 0, 0)$. Zangwill points out that when we choose $(\frac{1}{2}, 1, \frac{1}{2})$ as the initial point, the optimal point $(0, 0, 0)$ can never be reached, even in the limit, by using Powell's original procedure [8]. Essentially, this is because after performing the univariate search along each coordinate direction, the x component of the new set of directions is zero, so that the x component can never be changed again. Moreover, he observes that this function can also be used as a counter-example to Powell's modified procedure (see section 2.3) when the starting point $(100, -1, \frac{5}{2})$ is chosen. He points out that after minimization first along the x direction and then along the y and z directions the point $(\frac{1}{2}, 1, \frac{1}{2})$ will be reached and the same problem is encountered. Accordingly Zangwill suggests the following modification to Powell's method.

Let e_r, $r = 1,2,\ldots,n$ be the normalized coordinate directions in E^n. Initially, let $x_0 \in E^n$ be the starting point, ϵ the required accuracy and $d_1^1, d_2^1, \ldots, d_n^1$ be given normalized directions in E^n.

Step (0): Calculate λ to minimize $f(x_0 + \lambda d_n^1)$ and define $x_0 = x_0 + \lambda d_n^1$. Set $t = 1$ and $k = 1$.

An iteration of the procedure during the k^{th} stage is as follows:

Step (i): Find α to minimize $f(x_0 + \alpha e_t)$. Set

$$t = \begin{cases} t+1 & \text{if } 1 < t < n \\ 1 & \text{if } t = n. \end{cases}$$

Step (ii): If $\alpha \neq 0$, let $x_0 = x_0 + \alpha e_t$; go to (iii). If $\alpha = 0$, repeat (i); if step (i) has been repeated n times in succession, go to (v).

Step (iii): For $r = 1,2,\ldots,n$ calculate λ_r to minimize $f(x_{r-1} + \lambda_r d_r^k)$ and define $x_r = x_{r-1} + \lambda_r d_r^k$.

Step (iv): Define $d_{n+1} = (x_n - x_0)/\|x_n - x_0\|$. Determine λ to minimize $f(x_n + \lambda d_{n+1})$. Set $x_0 = x_n + \lambda d_{n+1}$ and $d_r^{k+1} = d_{r+1}^k$, $r = 1,2,\ldots,n$. Repeat (i) with $k = k+1$.

Step (v): Stop. The point x_0 is the desired optimum.

Zangwill has shown that this procedure converges in a finite number of iterations whenever it is applied to a quadratic function with a positive definite Hessian matrix. Moreover, theoretical convergence is also established for a strictly convex continuously differentiable function. However, there are no practical results available to compare this modified method to Powell's algorithm.

Some remarks will be made here concerning the counterexample proposed by Zangwill. There are two FORTRAN subroutines available for Powell's modified direct search method; one is programmed by Powell called VA04A, the other is programmed by Dr. C. Chan, called MINIMI. Following some discussion of Zangwill's counter-example, Dr. C. Chan ran this example using both VA04A and MINIMI. He has shown that the optimal point (0, 0, 0) can be reached by applying both these two subroutines. Thus, it turns out that there is some misunderstanding on this point. We have therefore

not implemented Zangwill's modifications (as had been our original intention), but some independent work is in progress in this direction.

Some difficulties are mentioned in section 2.2 concerning the univariate search suggested by Powell, and these difficulties cause his algorithm to fail when finding new directions during the search. To overcome this, we prefer to try the following two-stage process.

Phase I: Apply some linear search technique, e.g. Davis, Swann and Campey [4], to bracket the minimum within a small interval.

Phase II: Locate the minimum in this interval by using the Fibonnaci search [4], or a search by Golden Section [4], within some prescribed accuracy.

As we can see, this combined linear search technique will be very effective if the function to be minimized is unimodal within a small neighborhood containing the minimum. In fact, if the objective function involves several parameters, but is nonlinear in only one parameter, this type of algorithm has been proved to be extremely effective and widely applicable, see Barrodale, Roberts and Hunt [3]. Indeed, if this two-stage linear search technique is combined with Powell's direct search algorithm, it might yield an even more satisfactory direct search method. However, numerical results are needed to enable an assessment to be made on this conjecture.

2.5. Powell's Generalized Least-Squares Method.

The generalized least-squares problem is to minimize a sum of squares of nonlinear functions. That is, it is required to find $x = (x_1, x_2, \ldots, x_n) \in E^n$ to minimize

$$F(x) = \sum_{k=1}^{m} \{f_k(x)\}^2 \ , \ m \geq n \ . \qquad (2.3)$$

If the actual minimum of $F(x)$ is at $x + \delta$ then, by differentiating (2.3),

$$\sum_{k=1}^{m} [g_{ki}(x + \delta)\ f_k(x + \delta)] = 0 \ , \ i = 1,2,\ldots,n. \qquad (2.4)$$

where $g_{ki}(x + \delta) = \frac{\partial}{\partial x_i} f_k(x + \delta)$. By approximating the left-hand side of (2.4) by the first two terms of the Taylor series in δ about x, we obtain:

$$\sum_{k=1}^{m} \left[g_{ki}(x) + f_k(x) + \sum_{j=1}^{n} \{G_{ij}^{(k)}(x)\ f_k(x) + g_{ki}(x)\ g_{kj}(x)\} \right] \delta_j \approx 0. \qquad (2.5)$$

where $G_{ij}^{(k)}(x) = \frac{\partial^2}{\partial x_i \partial x_j} f_k(x)$. The increment vector δ can then be solved from (2.5).

The method proposed by Powell [9] to find the minimum of the function defined in (2.3) does not require any derivatives, and, in fact, it chooses linearly independent directions of search and approximates the first partial derivatives by differences.

Initially, let $y = (y_1, y_2,\ldots,y_n)$ be the initial estimate of the minimum and $\varepsilon > 0$ be the required accuracy. The initial step is to calculate approximate first partial derivatives. To do this, it requires increments $\varepsilon_1, \varepsilon_2,\ldots, \varepsilon_n$ to be specified which will yield reasonable estimates. They are given by:

$$\gamma_{ki} = \frac{f_k(y_1, y_2,\ldots,y_{i-1}, y_i + \varepsilon_i, y_{i+1},\ldots,y_n) - f_k(y)}{\varepsilon_i} \ , \ i = 1,2,\ldots,n.$$

For $i = 1,2,\ldots,n$, define

$$s_i = 1/\left\{ \sum_{k=1}^{m} \gamma_{ki}^2 \right\}^{\frac{1}{2}} .$$

Choose direction vectors d_i, with all components equal to zero except the i^{th} component which is equal to s_i, $i = 1,2,\ldots,n$. For $k = 1,2,\ldots,n$, define $\gamma_k = (\gamma_{k1}, \gamma_{k2}, \ldots, \gamma_{kn})$. Then the estimated derivative of the k^{th} function along the i^{th} direction is given by:

$$g_k(d_i) = \gamma_k \cdot d_i \quad , \quad i = 1,2,\ldots,n.$$

Note that the direction vector should be scaled so that

$$\sum_{k=1}^{m} [\gamma_{ki}]^2 = 1 \quad , \quad i = 1,2,\ldots,n.$$

The procedure for each complete iteration involves the following steps:

Step (1). Calculate the displacement vector δ:

Define

$$p_i = - \sum_{k=1}^{m} \gamma_{ki} f_k(y) \quad , \quad i = 1,2,\ldots,n.$$

Solve the following system of linear equations for g_js:

$$\sum_{j=1}^{n} \left[\sum_{k=1}^{m} \gamma_{ki} \gamma_{kj} \right] g_j = p_i \quad , \quad i = 1,2,\ldots,n.$$

The displacement vector δ is then given by:

$$\delta = \sum_{i=1}^{n} q_i d_i .$$

Step (2). Perform a linear search to find a minimum:

By using some linear search technique, find λ_m to minimize $F(y + \lambda\delta)$. Moreover determine λ_1 and λ_2 such that function values $f_k(y + \lambda_1\delta)$ and $f_k(y + \lambda_2\delta)$, $k = 1,2,\ldots,m$, yield the lowest and next lowest values of $F(y + \lambda\delta)$.

Step (3). Estimate the derivatives of the functions f_k in the direction δ:

The approximation is first predicted by:

$$u_k(\delta) = \frac{f_k(y + \lambda_1\delta) - f_k(y + \lambda_2\delta)}{(\lambda_1 - \lambda_2)}$$

Define

$$\mu = \sum_{k=1}^{m} \left[u_k(\delta) f_k(y + \lambda_m\delta) \right] / \sum_{k=1}^{m} \left[f_k(y + \lambda_m\delta) \right]^2$$

Then (2.6) is improved by:

$$g_k(\delta) = u_k(\delta) - \mu\, f_k(y + \lambda_m\delta)$$

Finally the vectors $g(\delta)$ and δ are scaled:

$$g_k(\delta) = g_k(\delta)/\| g(\delta)\| \quad \text{and} \quad \delta = \delta/\| g(\delta)\|$$

Where $\| g(\delta) \| = \left\{ \sum_{k=1}^{m} \left[g_k(\delta) \right]^2 \right\}^{\frac{1}{2}}$

Step (4). Replace one of the direction vectors $d_1, d_2, \ldots, d_n$ by δ:

Determine the integer t such that

$$|p_t q_t| = \max_{1 \leq i \leq n} |p_i q_i| \qquad (2.6)$$

Replace d_t by δ and set $y = y + \lambda_m\delta$.

Step (5). Test for Convergence:

Convergence is assumed to have occurred if both δ and $\lambda_m\delta$

have acceptably small components. Otherwise the same process is repeated from step (i).

Notice that if p_t is zero in (2.6), then the direction vector d_t cannot be replaced, until an iteration is started from a point different from the current y. This is to ensure that the directions $d_1, d_2, \ldots, d_n$ remain linearly independent.

Powell pointed out in his paper that the term $G_{ij}^{(k)}(x)\, f_k(x)$ in equation (2.5) can be ignored, because this term is of order δ if $f_k(x)$ is zero at the minimum, and it vanishes if f_k is linear in the variables. However, in practice, when a nonlinear function is fitted to a wide set of data, the terms f_k will no longer be small. In this situation, the accuracy of approximation will be lost because of truncating the second order terms in the Taylor series expansion.

CHAPTER THREE

GRADIENT METHODS

3.1. Introduction.

Gradient methods were defined earlier as those methods in which derivatives of the objective function are used in selecting the search direction. Thus, in this chapter, the objective function is assumed to be a real-valued differentiable function.

Let $f(x)$ be defined in E^n and let $x^{(0)} \in E^n$ be an initial estimate of the position of the required minimum. Then the general iterative rule of gradient methods is

$$x^{(i+1)} = x^{(i)} - \alpha_i H^{(i)} g^{(i)} , \quad i = 0,1,2,... \qquad (3.1)$$

where $g^{(i)}$ is the gradient vector of the objective function at $x = x^{(i)}$, α_i is a scalar, and $H^{(i)}$ is an $n \times n$ matrix which depends on $x^{(i)}$. Various gradient methods differ from each other in the choice of α_i and $H^{(i)}$ in (3.1). The following are commonly considered desirable conditions on α_i and $H^{(i)}$:

(A) α_i should be chosen in such a way that the following inequality holds:

$$f(x^{(i+1)}) \leq f(x^{(i)}) \qquad (3.2)$$

(B) $H^{(i)}$ should be some approximation to the inverse of the Hessian matrix of the objective function evaluated at the point $x^{(i)}$. For example, when $H^{(i)}$ is equal to the inverse of the Hessian matrix of $f(x)$ evaluated at $x^{(i)}$, and $\alpha_i = 1$, equation (3.1) is known as the Newton-Raphson method (see section 3.3).

In the following sections, other appropriate forms of α_i and $H^{(i)}$ will be considered.

3.2. The Steepest Descent Method.

If we choose the matrix $H^{(i)}$ in equation (3.1) to be the unit matrix I, then we have:

$$x^{(i+1)} = x^{(i)} - \alpha_i g^{(i)} , \tag{3.3}$$

which is known as the steepest descent method. The step-length α_i in (3.3) is often calculated to minimize the following function of one variable:

$$\phi(\alpha) = f(x^{(i)} + \alpha g^{(i)}) \tag{3.4}$$

The search for α_i in (3.4) is equivalent to a univariate search of section 2.2.

The rate of convergence of the steepest descent method which has been experienced by many computer users is very slow. For this reason, this method is not considered again in this thesis; the interested reader can consult [4, p. 34] for further details. However, the poor performance of this method was a strong motivation in the development of some of the more recent techniques.

3.3. The Newton-Raphson Method.

The Newton-Raphson method is a classical method, but it is still an excellent way of solving certain problems. Its special feature is that it makes use of second partial derivatives of the objective function.

Let $f(x)$ be a real-valued function defined in E^n. Assume that the second partial derivatives of $f(x)$ exist everywhere in E^n. The Newton-Raphson method estimates the position of the minimum of $f(x)$ from second and lower order terms in the Taylor series expansion. Consider the approximation:

$$f(x + \delta) \approx f(x) + \delta \cdot g + \frac{1}{2}\, \delta \cdot G \cdot \delta \ , \qquad (3.5)$$

where g is the gradient vector and G is the Hessian matrix of the function $f(x)$ evaluated at the point x.

Differentiating (3.5) and ignoring high order terms, gives:

$$\left[\frac{\partial f}{\partial x_i}\right]_{x+\delta} = \left[\frac{\partial f}{\partial x_i}\right]_x + \sum_{j=1}^{n} \delta_j \left[\frac{\partial^2 f}{\partial x_i\, \partial x_j}\right]_x \ , \quad i = 1,2,\ldots,n \qquad (3.6)$$

If $x + \delta$ is the required minimum of $f(x)$, then

$$\left[\frac{\partial f}{\partial x_i}\right]_{x+\delta} = 0 \ , \qquad i = 1,2,\ldots,n \ .$$

Thus we write (3.6) in the following form:

$$g + G \cdot \delta = 0$$

Solving the above linear system of equations for δ, yields:

$$\delta = -(G)^{-1}\, g \qquad (3.7)$$

Accordingly, the iterative formula for the Newton-Raphson method may be written as:

$$x^{(i+1)} = x^{(i)} - (G^{(i)})^{-1} \cdot g^{(i)} \qquad i = 0,1,2,\ldots \ . \qquad (3.8)$$

Here, $G^{(i)}$ and $g^{(i)}$ are the Hessian matrix and the gradient vector, respectively, of the function $f(x)$, both evaluated at the point $x^{(i)}$ and $x^{(0)}$ is initially chosen as some estimate of the required minimum.

Convergence is said to have occurred if, ultimately, for some i,

$$|x^{(i+1)} - x^{(i)}| < \varepsilon$$

where ε is the required accuracy in each component of x.

Some difficulties arise when using this method. First, if the initial estimate of the required minimum is poor, then the method often fails to converge. Secondly, even when it does converge, it may locate a point which is not a minimum, since the derivation of the iteration (3.8) depends only on the fact that the first derivatives of the objective function $f(x)$ are zero. However, as we pointed out in section 1.4., the first derivatives vanish at all stationary points of the objective function.

To avoid convergence to a stationary point that is not a minimum, the method may be modified as outlined in the next section.

3.4. The Modified Newton-Raphson Method.

Consider the following function of one variable:

$$\phi(\alpha) = f(x^{(i)} + \alpha\delta^{(i)}) \tag{3.9}$$

where $\delta^{(i)} = -(G^{(i)})^{-1}.g^{(i)}$. Let α_i be the minimum of $\phi(\alpha)$. Then the Newton Raphson method may be modified as follows:

$$x^{(i+1)} = x^{(i)} - \alpha_i(G^{i})^{-1}.g^{(i)}, \quad i = 0,1,2,... \tag{3.10}$$

The minimum of the function $\phi(\alpha)$ defined by (3.9), which we denote by α_i, may be obtained by applying some univariate search technique. The one we use in this thesis is that proposed by Davidson [4, p. 38]. He adopts a cubic interpolation scheme, using function values and gradients at two points, as follows. Let $x^{(i)}$, $\delta^{(i)}$, $g^{(i)}$ be given, and f^* be an estimate of the minimum of the function.

Step (i). Find another point x_2:

Evaluate $f_0 = f(x^{(i)})$ and find

$$\lambda = \min\left\{2, \frac{-2(f_0 - f^*)}{\delta^{(i)} \cdot g^{(i)}}\right\}$$

Let $x_2 = x^{(i)} + \lambda \cdot \delta^{(i)}$.

Calculate the function f_1 and the gradient vector g_1 at the point x_2 .

Step (ii). Test for convergence:

If $f_1 \geq f_0$ or $\delta^{(i)} \cdot g_1 \geq 0$, then the minimum has been straddled. A cubic is then fitted through these two points and the minimum of this cubic is found as an estimate of the minimum along a line.

Calculate:

$$z = \frac{3}{\lambda}(f_0 - f_1) + \delta^{(i)} \cdot g^{(i)} + \delta^{(i)} \cdot g_1$$

$$w = [z^2 - (\delta^{(i)} \cdot g^{(i)})(\delta^{(i)} \cdot g_1)]^{\frac{1}{2}}$$

$$\lambda^* = \lambda\left[1 - \frac{\delta^{(i)} \cdot g_1 + w - z}{\delta^{(i)} \cdot g_1 - \delta^{(i)} \cdot g^{(i)} + 2w}\right]$$

The minimum of the cubic is then given by $x^* = x^{(i)} + \lambda^* \delta^{(i)}$. An iteration of this linear search is complete. If the minimum has not been straddled, then go to step (iii).

Step (iii). Set

$$x^{(i)} = x^{(i)} + \lambda\ \delta^{(i)}$$

The process is repeated from step (i).

Consequently, we obtain an algorithm (3.10) that has quadratic convergence, and the ability to converge quite frequently from poor initial guesses

to the required solution. A suggested convergence criterion is that either the lengths, or every component, of the vectors $-[(G^{(i)})^{-1}.g^{(i)}]$ and $-[\alpha_i(G^{(i)})^{-1}.g^{(i)}]$ be less than the required accuracy.

The modified Newton-Raphson method has been used successfully in practice. However, the computation of the Hessian matrix $G^{(i)}$ and the determination of its inverse $(G^i)^{-1}$ in (3.10) are often very time consuming operations. To many computer users, the most serious disadvantage of this method is that it requires second partial derivatives of the objective function. Therefore techniques have been developed to attain fast convergence without the explicit evaluation of second partial derivatives.

3.5. Marquardt's Least-Squares Method.

Marquardt [6] has developed an algorithm for the least-squares estimation of nonlinear parameters. In his method, only function values and values of first partial derivatives are required. The method is based upon expanding the objective function in a Taylor series up to linear terms.

Consider the function $F(x,P)$, where $P = \{p_1, p_2,\ldots, p_n\}$ is a set of parameters and $x \in E^m$ is a vector which consists of m independent variables. Without loss of generality, we may assume that $m = 1$. Let $\{(x_i,y_i): i = 1,2,\ldots,N\}$ be a given set of data points. Suppose $F(x,P)$ is the function to be fitted to this data. The least-squares problem is to find $p_1, p_2,\ldots, p_n$ to minimize the function

$$f(P) = \sum_{i=1}^{N} \{y_i - F(x_i,P)\}^2 . \qquad (3.11)$$

Writing the Taylor series for $F(x,P)$ through linear terms:

$$F(x, P + \delta) \tilde{\sim} F(x,P) + g\cdot\delta = \hat{F}(x,P) \tag{3.12}$$

where g is the gradient vector of $F(x,P)$ with respect to P and δ is a vector in E^n. The value of $f(P + \delta)$ predicted by (3.2) is then given by:

$$\hat{f}(P) = \sum_{i=1}^{N} \{y_i - F(x_i,P) - g\cdot\delta\}^2 \tag{3.13}$$

If $P + \delta$ is a minimum of $f(P)$ defined in (3.11), then we have:

$$\left.\frac{\partial f}{\partial p_i}\right|_{p+\delta} = 0 , \quad i = 1,2,\ldots,n .$$

Differentiating (3.13) with respect to $p_1, p_2, \ldots, p_n$ successively and setting the resultant functions equal to zero, yields

$$G\cdot\delta = b \tag{3.14}$$

where $G = g\cdot g$ is an $n \times n$ matrix and b is an $n \times 1$ vector consisting of the following components:

$$b_j = \sum_{i=1}^{N} (y_i - F(x_i,P)) \frac{\partial F}{\partial p_j} , \quad j = 1,2,\ldots,n .$$

Instead of solving (3.14) for δ directly, the following extended system of linear equations have been considered:

$$(G + \lambda I)\cdot\delta = b \tag{3.15}$$

where I is the identity matrix and λ is a scalar satisfying certain conditions. In Marquardt's algorithm, elements of the matrix G and elements of the vectors b and δ have to be scaled as follows:

$$g_{ij} = \frac{g_{ij}}{\sqrt{g_{ii}}\sqrt{g_{jj}}} \qquad i,j = 1,2,\ldots,n$$

$$b_j = \frac{b_j}{\sqrt{g_{ij}}} \qquad j = 1,2,\ldots,n$$

$$\delta_j = \frac{\delta_j}{\sqrt{g_{jj}}} \qquad j = 1,2,\ldots,n .$$

Let $P^{(0)} = \{p_1^{(0)}, p_2^{(0)}, \ldots, p_n^{(0)}\}$ be the initial estimate of the minimum of the function $f(P)$ defined in (3.11). At the k^{th} iteration, the equation

$$(G^{(k)} + \lambda_k I) \cdot \delta^{(k)} = b^{(k)} \tag{3.16}$$

is constructed. This equation is then solved for $\delta^{(k)}$ and scaled. Set $P^{(k+1)} = P^{(k)} + \delta^{(k)}$ and compute $f(P^{(k+1)})$. It is desired to choose λ_k in (3.16) to maximize $[f(P^{(k)}) - f(P^{(k+1)})]$. Thus, before λ_k has been determined, f can be considered as a function of λ. The following trial and error strategy was suggested by Marquardt to find the optimal value of λ:

Initially, choose $\lambda_0 = 10^{-2}$. Let λ_{k-1} be the optimal value of λ from the previous iteration and $r = 10$. Define $f_1 = f(\lambda_{k-1})$, $f_2 = f(\lambda_{k-1}/r)$ and $f_3 = f(P^{(k)})$.

(i) If $f_2 \leq f_3$, let $\lambda_k = \lambda_{k-1}/r$

(ii) If $f_2 > f_3$ and $f_1 \leq f_3$, let $\lambda_k = \lambda_{k-1}$

(iii) If $f_2 > f_3$ and $f_1 > f_3$, increase λ by successive multiplication by r until for some smallest s,

$$f(\lambda_{k-1} \cdot r^s) \leq f_3 \ .$$

Then set $\lambda_k = \lambda_{k-1} r^s$. For further details of this strategy, one can consult [6].

The convergence criteria suggested by Marquardt is that the iterations should cease when

$$\frac{\delta_j^{(k)}}{\tau + |b_j^{(k)}|} < \varepsilon \qquad j = 1,2,\ldots,n$$

for some suitable small number τ, say 10^{-3}, and ε is the required accuracy.

The weakness of this algorithm is that it fails to provide a satisfactory strategy for choosing the optimal value of λ in (3.16). During the choice of λ the objective function has to be evaluated and the system of linear equations has to be solved for each trial value of λ, and these are time-consuming operations. Computational experience with this algorithm will be discussed in Chapter 5.

3.6. Powell's Method Requiring First Derivatives.

Powell [10] and [11] recently developed a new algorithm for calculating the minimum of a real-valued differentiable objective function $f(x)$ defined in E^n. In his algorithm, revised approximations to the Hessian matrix and its inverse are used. These new formulas can provide accurate second derivative approximations, even when the true Hessian matrix is not positive definite. Note that $\| \cdot \|$ will stand for the usual Euclidean norm in E^n .

Let $x^{(0)}$ be the initial estimate of the position of the required minimum and $\Delta^{(0)}$ the initial upper bound on the step length. The k^{th} iteration of the procedure calculates $x^{(k+1)}$ from $x^{(k)}$. Consider the approximation to $f(x^{(k)} + \delta)$:

$$f(x^{(k)} + \delta) \approx f(x^{(k)}) + \delta \cdot g^{(k)} + \frac{1}{2}\, \delta \cdot G^{(k)} \cdot \delta \qquad (3.17)$$

where $g^{(k)}$ and $G^{(k)}$ are the gradient vector and the Hessian matrix of $f(x)$, respectively, evaluated at $x = x^{(k)}$. $G^{(0)}$ and its inverse $H^{(0)}$ are set to λI and $\lambda^{-1} I$, respectively where I is the unit matrix, and λ is the scalar $0.01\ \|g^{(0)}\|/\Delta^{(0)}$.

It is desired to calculate a displacement vector $\delta^{(k)}$ from (3.17) satisfying the conditions:

$$\|\delta^{(k)}\| \leq \Delta^{(k)} \quad \text{and} \quad f(x^{(k)} + \delta^{(k)}) < f(x^{(k)}) .$$

To calculate $\delta^{(k)}$, the way suggested by Marquardt (see last section) is followed, but the number of computer operations has been reduced by a multiple of n. In fact, $\delta^{(k)}$ is forced to have the form

$$\delta^{(k)} = \alpha\, g^{(k)} + \beta [G^{(k)}]^{-1} \cdot g^{(k)}$$

where α and β are parameters. The following strategy is used by Powell for choosing α and β :

(i) If $(g^{(k)} \cdot G^{(k)} \cdot g^{(k)})\, \Delta^{(k)} \leq \|g^{(k)}\|^3$, then set $\alpha = -\Delta^{(k)}/\|g^{(k)}\|$ and $\beta = 0$; otherwise go to (ii).

(ii) First, calculate the following vectors:

$$v^{(k)} = -H^{(k)} \cdot g^{(k)}$$

and $$s^{(k)} = -\|g^{(k)}\|^2\, g^{(k)} / (g^{(k)} \cdot G^{(k)} \cdot g^{(k)})$$

Then $\delta^{(k)}$ is given by:

$$\delta^{(k)} = s^{(k)} + \theta^* \{v^{(k)} - s^{(k)}\}$$

where $\theta^* = \min\{1, \theta\}$

and $$\theta = \frac{\{\Delta^{(k)2} - \|s^{(k)}\|^2\}\sin(s^{(k)}, w^{(k)})}{|s^{(k)}\cdot w^{(k)}| + \{(s^{(k)}\cdot w^{(k)})^2 + \|w^{(k)}\|^2[(\Delta^{(k)})^2 - \|s^{(k)}\|^2]\}^{\frac{1}{2}}}$$

where $w^{(k)} = v^{(k)} - s^{(k)}$.

After finding $\delta^{(k)}$, define

$$\gamma^{(k)} = g(x^{(k)} + \delta^{(k)}) - g^{(k)} \tag{3.18}$$

and $$x^{(k+1)} = \begin{cases} x^{(k)} & \text{, if } f(x^{(k)} + \delta^{(k)}) \geq f(x^{(k)}) \\ x^{(k)} + \delta^{(k)} & \text{, otherwise .} \end{cases}$$

Convergence is assumed if $\|g^{(k+1)}\| \leq \varepsilon$, where ε is the required accuracy.

The calculation of $G^{(k+1)}$ and $H^{(k+1)}$ is considered now. The revised formulas proposed by Powell are as follows:

$$G^* = G + \frac{\mu\cdot\delta^T + \delta\cdot\mu^T}{\|\delta\|^2} - \frac{\delta\cdot\delta^T(\mu^T\cdot\delta)}{\|\delta\|^4} \tag{3.19}$$

$$H^* = H - \{\eta\cdot\eta^T(\delta^T\cdot H\cdot\delta) - [\eta\cdot\delta^T\cdot H + H\cdot\delta\cdot\eta^T](\delta^T\cdot H\cdot\gamma) + H\cdot\delta\cdot\delta^T\cdot H(\eta^T\cdot\gamma)\}/\{(\eta^T\cdot\gamma)(\delta^T\cdot H\cdot\delta) - (\delta^T\cdot H\cdot\gamma)^2\} \tag{3.20}$$

where

$$\mu = \gamma - G\cdot\delta \ , \quad \eta = H\cdot\gamma - \delta$$

and the superscript "T" denotes the transpose.

Usually, we let $\delta = \delta^{(k)}$, $\gamma = \gamma^{(k)}$, $G = G^{(k)}$, $G^{(k+1)} = G^*$, $H = H^{(k)}$ and $H^{(k+1)} = H^*$. We note that equation (3.20) will fail if the denominator is zero; this happens when $G^{(k+1)}$ is singular. To avoid singularity, replace (3.19) by:

$$G^{*}(\theta) = G + \theta \frac{\mu \cdot \delta^{T} + \delta \cdot \mu^{T}}{\|\delta\|^{2}} - \theta^{2} \frac{\delta \cdot \delta^{T} (\mu^{T} \cdot \delta)}{\|\delta\|^{4}} \tag{3.21}$$

It is desired to find a suitable value of θ, say $\theta^{(k)}$, as close to 1 as possible and having the property that

$$|\det G^{(k+1)}| \geq 0.1 \, |\det G^{(k)}| \tag{3.22}$$

The special formula (3.21) is needed to define $G^{(k+1)}$ in order to satisfy inequality (3.22) if the inequality

$$|(\delta^{T} \cdot H \cdot \gamma)^{2} - (\delta^{T} \cdot H \cdot \delta)(\eta^{T} \cdot \gamma)| < 0.1 \, \|\delta\|^{4} \tag{3.23}$$

holds. Powell proved that $G^{*}(\theta)$ in (3.21) is singular if and only if θ satisfies the quadratic equation:

$$0.9 \, \|\delta\|^{4} + 2\theta \|\delta\|^{2} (\delta^{T} \cdot H \cdot \mu) + \theta^{2} \{(\delta^{T} \cdot H \cdot \mu)^{2} - (\delta^{T} \cdot H \cdot \delta)(\mu^{T} \cdot H \cdot \mu) - (\delta^{T} \cdot H \cdot \delta)(\delta^{T} \cdot \mu)\} = 0 \tag{3.24}$$

To find $\theta^{(k)}$, define $\phi = 1 - \theta$ and

$$\sigma = (\eta^{T} \cdot \gamma)(\delta^{T} \cdot H \cdot \delta) - (\gamma^{T} \cdot H \cdot \delta)^{2} \tag{3.25}$$

Then ϕ is given by:

$$\frac{\sigma + 0.1 \, \|\delta\|^{4}}{\{\sigma + (\delta^{T} \cdot H \cdot \gamma) \, \|\delta\|^{2}\} \pm \|\delta\|^{2} \left[\{\delta^{T} H \gamma - 0.1 \, \|\delta\|^{2}\}^{2} + 0.9\{\sigma + 0.1 \, \|\delta\|^{4}\}\right]^{\frac{1}{2}}} \tag{3.26}$$

Since we require $\theta^{(k)}$ to be close to 1, the zero that is smaller in modulus, say $\phi^{(k)}$, in the above expression (3.26) is chosen. Hence we let the sign of expression (3.26) be the same as the sign of $\{\sigma + (\delta^{T} \cdot H \cdot \gamma) \, \|\delta\|^{2}\}$.

Thus we obtain $\theta^{(k)} = 1 - \phi^{(k)}$.

Instead of calculating $G^{(k+1)}$ by substituting $G = G^{(k)}$, $\delta = \delta^{(k)}$, $\mu = \gamma^{(k)} - G^{(k)}.\delta^{(k)}$ and $\theta = \theta^{(k)}$ in expression (3.21), one may obtain the same matrix by setting $G = G^{(k)}$, $\delta = \delta^{(k)}$ and

$$\gamma = \theta^{(k)}\gamma^{(k)} + (1-\theta^{(k)})\, G^{(k)}.\delta^{(k)} + \{\theta^{(k)}(1-\theta^{(k)}) \\ (\gamma^{(k)} - G^{(k)}.\delta^{(k)})^T \cdot \delta^{(k)}\}\delta^{(k)}/\|\delta^{(k)}\|^2 \tag{3.27}$$

in expression (3.19). This method is preferred because it allows us to use equation (3.20) to define $H^{(k+1)}$, where $H = H^{(k)}$, $\delta = \delta^{(k)}$ and γ is defined by (3.27).

The purpose of choosing the step-bound is to make $\| \delta \|$ so small that the approximation (3.17) is adequate for $\delta = \delta^{(k)}$. Three values for the step-bound $\Delta^{(k+1)}$ for the $(k+1)^{st}$ iteration may be chosen, according to the following strategy.

Consider the function with one parameter λ :

$$\psi(\lambda) = g(x^{(k)} + \lambda\, \delta^{(k)})^T.\delta^{(k)}$$

Powell showed that $f(x^{(k)} + \lambda\, \delta^{(k)})$ is least when λ is the number

$$\lambda^* = \begin{cases} \infty\ , & \psi(1) \le \psi(0) \\ \psi(0)/\{\psi(0) - \psi(1)\}, & \psi(1) > \psi(0)\ . \end{cases} \tag{3.28}$$

The following two testing inequalities are needed:

$$f(x^{(k)} + \delta^{(k)}) - f(x^{(k)}) \le 0.1\{g^{(k)}.\delta^{(k)} + \frac{1}{2}\, \delta^{(k)}.G^{(k)}.\delta^{(k)}\} \tag{3.29}$$

$$\|g(x^{(k)} + \delta^{(k)}) - g^{(k)} - G^{(k)}.\delta^{(k)}\|^2 \le \frac{1}{4}\, \|g^{(k)}\|^2 \tag{3.30}$$

Then we define:

$$\Delta^{(k+1)} = \begin{cases} \frac{1}{2}\Delta^{(k)}, & \text{if (3.29) failed.} \\ \Delta^{(k)}, & \text{if both (3.29) and (3.30) failed, and } \lambda^* < 2. \\ 2\Delta^{(k)}, & \text{otherwise.} \end{cases}$$

The stability of formulas (3.19) and (3.20) for calculating the Hessian matrix G and its inverse H need to be considered. Since both formulas are used iteratively, difficulty will arise due to computer rounding error. Small rounding error of a single iteration may be negligible, while after a number of iterations the total error accumulated by those small errors may be disastrous. Powell proved that these formulas are stable [11, p. 30]. Moreover, every third iteration is a "special iteration" in his algorithm in order to improve the approximation $G^{(k+1)}$. He pointed out that these special iterations are included, not only because they are needed to prove some convergence theorems, but also numerical experiments have indicated that they are worthwhile.

Convergence theorems together with some numerical results in the report indicate that the new method may be preferable to current algorithms in this class for solving some unconstrained minimization problems.

CHAPTER FOUR

THE RICHARDS FUNCTION

4.1. Introduction.

Let $\{(x_i, y_i) : i = 1,2,\ldots,N\}$ be a given set of data. The curve fitting problem is to first choose an approximating function $F(x,P)$, and then to find a particular form $F(x, P^*)$ which fits the given set of data satisfactorily. Here, $P = \{p_1, p_2,\ldots, p_n\}$ is a set of free parameters and x is an independent variable. The function $F(x,P)$ is said to be linear if it depends linearly upon all of the parameters $p_1, p_2,\ldots, p_n$; otherwise it is nonlinear. $F(x, P^*)$ is called a best approximation in an ℓ_p norm if, for all choices of P, we have:

$$\|y - F(x, P^*)\|_p \leq \|y - F(x,P)\|_p ,$$

where
$$\|y - F(x,P)\|_p = \begin{cases} \left[\sum_{i=1}^{N} |y_i - F(x_i, P)|^p\right]^{1/p} , & 1 \leq p < \infty \\ \max_{1\leq i\leq N} |y_i - F(x_i, P)| , & p = \infty. \end{cases}$$

The three norms most commonly used in practice are ℓ_1, ℓ_2 and ℓ_∞. For fitting a curve, the particular choice of norm depends on whether the errors are widely inaccurate (ℓ_1), normally distributed (ℓ_2), or small (ℓ_∞) relative to the error of approximation (see Barrodale [1]). Best approximations exist in all three norms for linear approximations, but only ℓ_2 approximations are necessarily unique (see Rice [13] and [14]). Best ℓ_1 and ℓ_∞ linear approximations can be determined by linear programming (see Barrodale and Young [2]), while ℓ_2 (i.e. least-squares) linear approximations are determined by solving a system of linear algebraic equations, called the normal equations (see Rice [13]). If $F(x,P)$ is a function of several parameters which depends nonlinearly on just one of these parameters, then best approximations in all three norms can usually be computed satisfactorily, provided that a best approximation exists

(see Barrodale, Roberts and Hunt [3]). However, in general, best approximations do not necessarily exist for discrete nonlinear approximations (see Rice [14]). The development of new algorithms for best nonlinear approximations is an area of much current research activity in numerical analysis.

4.2. The Richards Function.

The Richards function is defined by the differential equation (see Causton [5])

$$\frac{dW}{dt} = \frac{kW}{nA^n}(A^n - W^n) , \qquad (4.1)$$

where W represents the weight or the size of an organism at time t, and A, k and n are parameters.

It can be verified that the integrated form of (4.1) is

$$W = A(1 \pm be^{-kt})^{-1/n} , \qquad (4.2)$$

where b is associated with the constant of integration, and the plus sign within the bracket is chosen when $n > 0$, while the negative sign is applicable when $-1 \leq n < 0$. The function is not defined for $n < -1$ or $n = 0$. There are some biological implications of these constants. For example, A is the ultimate limiting value of W, i.e. $\lim_{t\to\infty} W(t) = A$, and so it represents the final size of the plant or organism. The shapes of these growth curves are determined by the different values of n. Thus, when $n = -1$, equation (4.2) reduces to

$$W = A(1 - be^{-kt}) ,$$

which is the monomolecular curve; it has been used (see Richards [15])

to represent the later portions of life history. If $n = 1$, we have

$$W = A/(1 + be^{-kt}) ,$$

which is known as the logistic curve; it is used as a convenient empirical growth curve [15]. Since $\lim_{x\to 0} \frac{a^x - 1}{x} = \ell n\ a$, it follows from (4.1) that

$$\lim_{n\to 0^-} \frac{dW}{dt} = kW\ \ell n\ (A/W)$$

which is the growth-rate of the so-called Gompertz curve. Consequently, (4.2) approximates the Gompertz curve

$$W = A\cdot Exp\left[-be^{-kt}\right],$$

as $n \to 0^-$; this curve has been used in population studies and in representing animal growth [15].

As we can see, the Richards function includes as special cases several curves which have been used empirically for the description of growth. In fact, Causton [5] remarks that the Richards function is probably, as yet, "the most realistic mathematical description of plant and animal growth".

4.3. Previous Attempts.

Richards [15] describes an empirical method of fitting (4.2) to experimental data, which is based on the linear regression technique. Consider the linear form

$$y = \ell n\ B + Kt , \qquad (4.3)$$

where $y = \ell n\left|(A/W)^n - 1\right|$, $B = |b|$ and $K = -k$. Notice that equation (4.3)

can be obtained by rearranging terms in equation (4.2), and then taking logarithms. Suppose $\{(t_i, W_i) : i = 1,2,\ldots,N\}$ is a given set of experimental data. To calculate values of y_i, values of A and n must be first provided, the remaining two parameters B and k may then be found by linear regression. A convenient method of obtaining the starting values of A and n has been proposed by Richards. In fact, the value of A is chosen initially to be the maximum value of the W_i's. It can be shown that

$$W_I = A(n+1)^{-1/n} \tag{4.4}$$

corresponds to the point of inflection of the curve (4.2). Note that (4.4) is not defined for $n = -1$, because the monomolecular curve has no point of inflection. The starting value of n is now readily obtained from (4.4), where W_I is estimated from the graph of W_i against t_i. The approximations may then be improved by adjusting values of A or n in such a way that the deviations from linearity in (4.3) are small [15]. In practice, it has been found that difficulties will arise in determining the correct adjustments of A and n to give a straight line with (4.3), particularly when the given set of data is very irregular. In fact, Causton [5] pointed out that the Richards' method is very laborious, and sometimes it may produce misleading results.

A statistical method of fitting (4.2) by least-squares has been proposed by Nelder [7]. By taking natural logarithms, (4.2) reduces to

$$w = a + m \ln (1 + Be^{Kt}) \tag{4.5}$$

where $w = \ln W$, $a = \ln A$, $m = -\frac{1}{n}$, $B = |b|$ and $K = -k$. Nelder's method is based on solving the so-called normal equations, induced by fitting

(4.5) to experimental data, by using the generalized Newton-Raphson method [7]. His calculations were performed by hand, and he provides tables to assist in the computation. A semi-graphical method of obtaining starting values for the parameters a, m, B and K in (4.5) has been described in [7]. Causton [5] observes that Nelder's method is a time-consuming procedure.

Accordingly, Causton describes an analytical method of fitting (4.2) by least-squares. In his method, equation (4.5) is the form in which curves are fitted to the data. Thus, the problem is to minimize the function

$$f(a, m, B, K) = \sum_{i=1}^{N} \{w_i - a - m \ln (1 + Be^{Kt_i})\}^2 . \qquad (4.6)$$

Causton's method is based on solving the so-called normal equations of (4.6) by using the Newton-Raphson technique in four dimensions. Moreover, he proposed an analytical procedure to generate the starting values of the parameters A, n, b and k. In practice, it is our experience that this procedure does provide good starting values, and discussion of this procedure in detail will be considered in section 4.5. Consequently, Causton's method is probably the most efficient method among these previous attempts. However, as we shall see in the next section, this method can be improved further.

4.4. The New Approach.

Consider the problem of fitting the Richards function to experimental data. When the given set of data is irregular, solving the problem by using the ℓ_1 norm would perhaps be more meaningful. However, this is infeasible at this time, since no algorithm is available. The ℓ_2 norm is chosen because the objective function associated with this problem is

differentiable, and, as we shall see later, this enables us to handle the problem more satisfactorily in practice.

In view of the fact that the magnitude of the parameter B is large compared with the others in (4.6), we prefer to set $B = e^{\beta}$. Furthermore, before attempting any approximation, the time interval of the approximation is transformed from [1, 365] to [0,1]. This is done for consistency and ease of manipulation in the computation.

Accordingly, equation (4.6) becomes

$$w = a + m \ln (1 + e^{\beta+Kt}) , \tag{4.7}$$

and this is the form in which curves are fitted to the data. The objective function defined in (4.7) now reduces to

$$f(a, m, \beta, K) = \sum_{i=1}^{N} \{w_i - a - m \ln(1 + e^{\beta+Kt_i})\}^2 . \tag{4.8}$$

The so-called normal equations of (4.8) are given in Appendix I. Notice that the function f defined in (4.8) is linear in the parameters a and m, and it is nonlinear in β and K. This suggests that we are able to solve a and m in terms of β and K from the normal equations (A.2) and (A.3) given in Appendix I. The expressions $a = g_1(\beta,K)$ and $m = g_2(\beta,K)$ are given in equations (A.7)-(A.13) in Appendix I.

The objective function to be minimized thus becomes

$$f(\beta,K) = \sum_{i=1}^{N} \{w_i - g_1(\beta,K) - g_2(\beta,K) \ln(1 + e^{\beta+Kt_i})\}^2 , \tag{4.9}$$

after substituting the results obtained for a and m into equation (4.8). Observing (4.9), we note that f is now a function involving two nonlinear parameters β and K only. Consequently, the original 4 × 4 system associated

with (4.8) has been reduced to the 2 × 2 system obtained by deriving the normal equations for (4.9). We then proceed to solve this new system by applying some of the techniques described previously. The procedure for generating the starting values will be discussed in the next section. Elements of the gradient vector as well as that of the Hessian matrix of f defined in (4.9) are given in Appendix II. Furthermore, formulas for solving this new (2 × 2) system by the Newton-Raphson method are presented in the same Appendix.

In the next chapter we discuss briefly the computational savings that result from regarding this least-squares problem as a two-dimensional, rather than a four-dimensional, problem.

4.5. Calculating the Starting Values.

The method proposed by Causton [5] will be used for generating the starting values of the parameters A, n, B and k when the Richards function (4.2) is fitted to experimental data, say $\{(t_i, W_i) : i = 1,2,\ldots,N\}$. These four starting values are needed when solving the problem by minimizing the function defined in (4.8). However, if the new approach is applied, only the starting values of β and K are required.

The relative growth-rate of the curve (4.2) is defined by:

$$R = \frac{1}{W}\frac{dW}{dt} .$$

It follows from (4.1) that R can be written in the form

$$R = u + vW^n , \tag{4.10}$$

where

$$u = \frac{k}{n} \quad , \quad v = -\frac{k}{nA^n} \tag{4.11-4.12}$$

Thus, if (4.10) is fitted, the starting values of A, n and k are determined, and the starting value of B can then be obtained from (4.3) by linear regression. To fit (4.10), values of R must be calculated from the given set of data. In fact, the mean value of R over a period of time between t_1 and t_2, when the weights are W_1 and W_2 respectively, is given by (see [5]):

$$\overline{R} = \frac{\ell n\, W_2 - \ell n\, W_1}{t_2 - t_1} \tag{4.13}$$

According to (4.13), we compute:

$$R_{i+1} = (\ell n\, W_{i+2} - \ell n\, W_i)/(t_{i+2} - t_i)\ , \quad i = 1,2,\ldots,N-2, \tag{4.14}$$

corresponding to W_{i+1} for the purpose of fitting (4.10). Two end points W_1 and W_2 being, of necessity, omitted when (4.10) is fitted. The whole process of calculating the starting values is iterative on values of n, and has been summarized as follows:

Initial step. Set $n = -1$ and $r = 1$.

Compute R_{i+1}, $i = 1,2,\ldots,N-2$, by using (4.14).

Step (i). Find u and v by linear regression:

Define

$$\overline{R} = \sum_{i=2}^{N-1} R_i\ /(N-2)$$

$$\overline{W} = \sum_{i=2}^{N-1} W_i^n\ /(N-2)$$

Then u and v are given by:

$$v = \sum_{i=2}^{N-2} (R_i - \overline{R})(W_i^n - \overline{W}) / \sum_{i=2}^{N} (W_i^n - \overline{W})^2$$

$$u = \bar{R} - v\,\bar{W} .$$

Step (ii). Compute the sum of squares:

Define

$$\delta_r = \sum_{i=2}^{N-2} (R_i - u - v\,W_i^n)^2$$

If $r = 1$, then set $r = r+1$ and $n = n+0.1$; go to (i). Otherwise go to (iii).

Step (iii). Test for convergence:

If $\delta_r \leq \delta_{r-1}$ then set $r = r+1$ and $n = n+0.1$; go to (i). Otherwise convergence is assumed to have occurred, and the value of n obtained at this stage is optimal; go to (iv).

Step (iv). Compute the starting values of k and A by using (4.11) and (4.12). From (4.3), the value of B is then readily given by $\ell n\, B = \left\{ \sum_{i=1}^{N'} y_i + k \sum_{i=1}^{N'} t_i \right\} / N'$, where N' is the number of points remaining, when those in which $W_i \geq A$ have been eliminated.

Some comments will be noted here. First, since the curve (4.2) is not defined for $n = 0$, a test for when this situation occurs must be inserted in step (iii). Secondly, the sign of k must be positive, for otherwise the value of A given in (4.12) will be undetermined. In fact, when n is positive, k will always be positive because the straight line given in (4.10) will be situated as shown in Figure 4.1. with $n > 0$. While if n lies in the range, $-1 \leq n < 0$ the line should occur as shown in Figure 4.2 with $u < 0$.

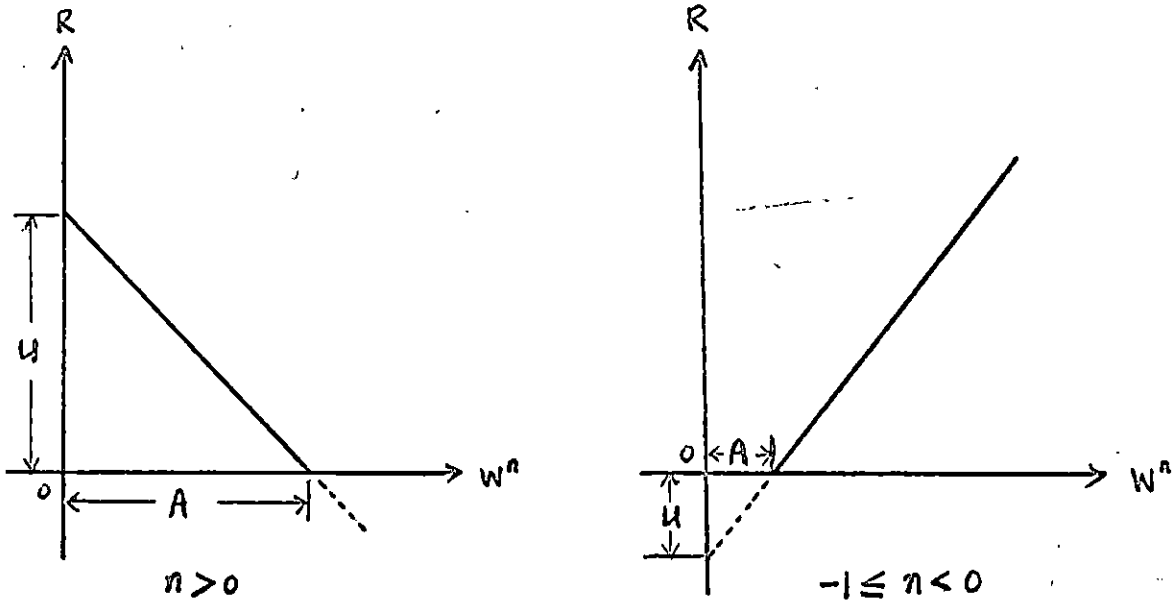

Figure 4.1. The linear relationship between W^n and R.

CHAPTER FIVE

NUMERICAL STUDIES

5.1. Introduction.

5.2. Experimental Data.

5.3. Choice of Algorithms.

5.4. Numerical Results.

5.5. Computational Remarks.

5.6. Comparisons and Conclusions.

5.1. Introduction.

In this chapter we present a summary of our numerical experience in fitting the Richards function to experimental data, using the methods described in the preceding chapters.

In tabulating the results, we have estimated the number of function evaluations as well as the computing times required by each algorithm. For methods which require derivatives, the function value and the partial derivatives with respect to each parameter are evaluated at each iteration. We assume that the computational effort involved in calculating each partial derivative can be equated with the function evaluations. Thus, for each method, the total number of function evaluations in each iteration is given by

$$n_f = n_2 + n_1 + \ell + 1 \tag{5.1}$$

where n_2 and n_1 are, respectively, the total number of evaluations of the second partial derivatives and the first partial derivatives, while ℓ is the number of function evaluations made during the linear search. Consequently, in comparing these techniques, the total number of function evaluations required for each method is given by:

$$n_t = n_i \cdot n_f \ , \tag{5.2}$$

where n_i is the number of iterations and n_f is defined by (5.1). The computing time required for each algorithm to solve a given problem is defined as the execution time (in seconds) involved; thus, compile times are ignored.

All the calculations have been carried out in double precision arithmetic on an IBM 360/44 computer.

5.2. Experimental Data.

Mean dry weight measurements were made on Douglas fir seedlings planted at 2, 4, 6 and 12 inch square spacings. We were given the results of these experiments, and we have tried to fit these data by the Richards function, using the least-squares method.

The data, denoted by ordered pairs (t_i, W_i), obtained from these four different spacings is presented in Table 5.1. Harvests were taken at 2-week intervals, except that the last sample was taken 28 days after the previous harvests. Figure 5.1 was constructed by taking natural logarithms of the W_i's, and plotting the results against t_i's.

These data were provided from experiments conducted by Dr. R. van den Driessche of the Research Division Laboratory, British Columbia Forest Service, Victoria, British Columbia. Details of these experiments are to appear in part in van den Driessche [16]. In short, his objective is to observe the growth differences in certain types of seedlings, planted at different spacings, in both fertilized and unfertilized plots of land. The suitability of the Richards function for describing these growths mathematically is also being investigated. Quite obviously, this requires that a suitable numerical technique for fitting the data by the Richards function be readily available.

TABLE 5.1.: Mean dry weight of Douglas fir (various spacings).

Time (Weeks)	2" × 2"	4" × 4"	6" × 6"	12" × 12"
2	.666	.547	.715	.553
4	.516	.570	.654	.575
6	.587	.721	.853	.733
8	.682	.749	.671	.667
10	.639	.922	1.010	.680
12	.845	1.261	1.103	.866
14	1.213	1.569	1.700	1.210
16	1.373	1.944	2.268	1.794
18	1.803	2.504	2.842	2.154
20	2.441	4.616	4.666	3.328
22	1.988	4.373	4.158	3.725
24	2.142	5.521	6.789	4.832
26	3.340	7.277	7.765	4.767
28	4.494	6.072	7.896	7.129
30	3.050	6.789	9.975	7.232
32	4.086	9.799	11.671	6.838
34	3.800	8.402	10.571	6.690
36	4.856	10.391	15.719	9.083
38	3.872	7.979	16.339	9.784
40	3.336	9.028	15.629	7.285
44	3.652	11.024	14.701	10.462

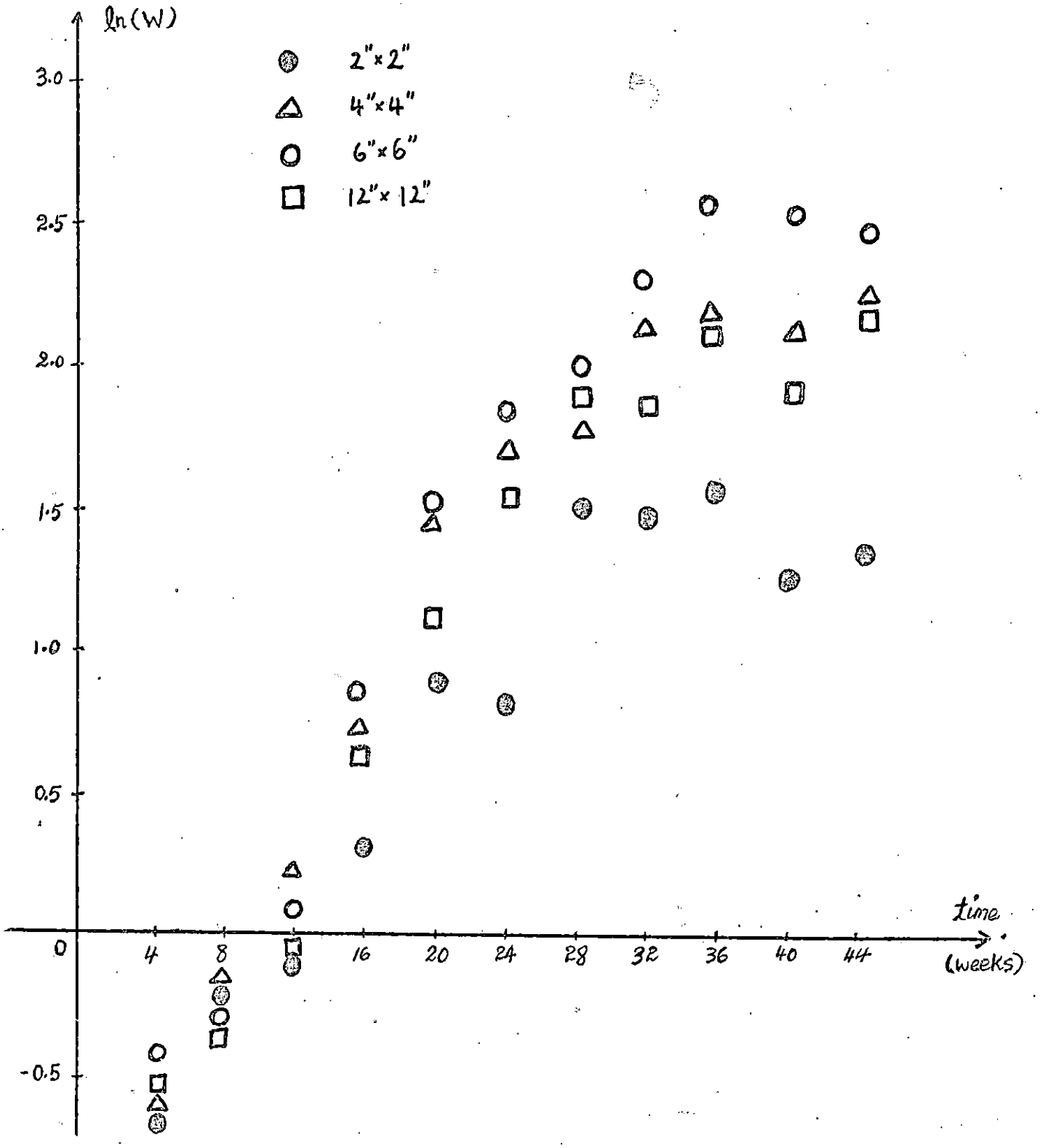

Figure 5.1.: The form in which the data in Table 5.1. is to be fitted.

5.3. Choice of Algorithms.

The algorithms used will be identified in the following manner:

(1) PDS - Powell's direct search method without calculating derivatives of the objective function, described in section 2.3.

(2) PGL2 - Powell's generalized least-squares technique, without calculating derivatives explicitly, described in section 2.5.

(3) PFD - Powell's method which requires the first partial derivatives; see section 3.6.

(4) ML2 - Marquardt's algorithm for least-squares curve fitting in which the function value and first partial derivatives are required; see section 3.5.

(5) NR - The Newton-Raphson method, described in section 3.3.

(6) NRD - The Newton-Raphson method with Davidson's linear search technique, described in section 3.4.

Dr. C. Chan has programmed PDS in FORTRAN IV in the form of a subroutine called MINIMI, and PGL2 in FORTRAN IV in the form of a main program called AP2(ND1). For ML2, a Share program named NLIN has been written in PL/I in the form of a main program. All of these three programs are available in the Computer Centre, University of Victoria. A FORTRAN subroutine for PFD is listed in the report given by Powell [11]. We have programmed NR and NRD (in the 2-dimensional case) in FORTRAN IV; these are listed in Appendices 3-7.

5.4. Numerical Results.

In this section, we present the numerical results obtained by fitting the Richards function to the experiment data given in Table 5.1 by using

the approach described in section 5.3 are used, and for each method the required accuracy has been chosen to be 10^{-3}.

For each set of data, the starting values of the parameters β and K in (4.8) are given in Table 5.2; these are obtained by applying the procedure described in section 4.5. Moreover, the results obtained by running each set of data with each one of the six algorithms are listed in Tables 5.3-5.6. In those tables, F^* denotes the final sum of squares defined in (4.8), β^* and K^* represent, respectively, the last values of the parameters β and K obtained by applying the six algorithms. The total number of function evaluations (n_t) required by the algorithms are also given. Ranks are assigned to these algorithms in the order of increasing n_t, and they are listed in the column called Rank I. Furthermore, the computing time (in seconds) required for each algorithm has been listed, and a rank is assigned to each algorithm in order of increasing amounts of computing time required. These ranks are given in the last column called Rank II in the corresponding tables.

TABLE 5.2.: The starting values of β and K.

Data set	β	K
2" × 2"	5.8644	-10.0485
4" × 4"	6.4400	-12.0300
6" × 6"	7.8674	-12.2916
12" × 12"	7.7723	-13.3742

TABLE 5.3.: Results for the first set of data. (2" × 2")

Algorithm	F^* (Sum of Squares)	β^*	K^*	n_t	Rank I	Computing time (secs.)	Rank II
NR	0.5910	79.3949	-156.1608	63	3	6.83	2
NRD	0.5918	68.7582	-134.9526	50	1	6.04	1
PFD	0.5905	87.8582	-173.1068	54	2	6.93	3
PDS	0.6023	19.0413	-36.8079	97	4	16.24	4
PGL2	-	-	-	-	6	-	6
ML2	0.7985	5.8644	-10.0485	2508**	5	88.12	5

** The need for a large number of function evaluations in ML2 is explained in Section 3.5.

TABLE 5.4.: Results for the second set of data. (4" × 4").

Algorithm	F^* (Sum of squares)	β^*	K^*	n_t	Rank I	Computing time (secs.)	Rank II
NR	0.4166	9.3971	-18.2741	28	2	5.62	1
NRD	0.4166	9.3994	-18.2788	30	3	6.15	3
PFD	0.4169	8.8853	-17.2940	27	1	6.10	2
PDS	0.4166	9.3999	-18.2796	51	4	11.49	4
PGL2	0.4166	9.4000	-18.2798	61	5	66.24	6
ML2	0.4569	6.4400	-12.0299	924	6	45.64	5

TABLE 5.5.: Results for the third set of data (6" × 6").

Algorithm	F^* (Sum of squares)	β^*	K^*	n_t	Rank I	Computing time (secs.)	Rank II
NR	0.6089	10.7879	-17.9657	21	2	5.39	1
NRD	0.6085	10.8451	-18.0529	20	1	5.72	2
PFD	0.6088	10.4018	-17.3246	30	3	6.07	3
PDS	0.6085	10.8927	-18.1391	62	4	13.22	4
PGL2	0.6085	10.8926	-18.1390	71	5	75.64	6
ML2	0.6584	7.8674	-12.2915	1470	6	60.95	5

TABLE 5.6.: Results for the fourth set of data (12" × 12").

Algorithm	F^* (Sum of squares)	β^*	K^*	n_t	Rank I	Computing time (secs.)	Rank II
NR	0.6448	11.2929	-20.3388	28	1	5.74	1
NRD	0.6448	11.4812	-20.6859	30	2	5.95	2
PFD	0.6450	10.9124	-19.6446	30	3	5.95	3
PDS	0.6448	11.4827	-20.6884	61	4	13.27	4
PGL2	0.6448	11.4827	-20.6884	79	5	84.28	6
ML2	0.6760	7.7723	-13.3742	987	6	47.44	5

5.5. Computational Remarks.

In practice, we have found that when the Richards function is fitted to experimental data which is very irregular, NR and NRD may fail to converge if the convergence criterion described in section 3.3 is used. Consequently, for the results given in Tables 5.3-5.6 the convergence criterion proposed by Powell (see section 3.5), i.e.

$$\| g^{(r)} \| \leq \varepsilon ,$$

is employed. Here $g^{(r)}$ is the gradient vector of the objective function (4.8) evaluated at the point $(\beta^{(r)}, K^{(r)})$ at the r^{th} iteration, and $\varepsilon = 10^{-3}$ is the required accuracy. When this convergence criterion is applied, both NR and NRD work more satisfactorily in practice.

From Figure 5.1, we note that the distribution of the first set of data (2" × 2") is very irregular. For this problem, almost every algorithm ended up with a different result (see Table 5.3). Indeed, PGL2 failed to converge. In fact, because of the widely separated distribution of the first set of data, in this case the error of approximation when using (4.8) is very insensitive to changes in its two parameters. As we can see from Table 5.3, the error of approximation (4.8) produces a flat function in a large area about its minimum.

From Tables 5.4-5.6, we note that best approximations are more sharply defined for these three sets of data, and the results obtained by all the algorithms, except ML2, are very close to each other. Notice that since

ML2, and the subroutines we supplied to it, are written in PL/I, it is difficult to make a comparison of the computing time required by this algorithm with those algorithms written in FORTRAN IV. Thus, the rank assigned

to ML2 in the Rank II column is not too meaningful.

5.6. Comparison and Conclusions.

In section 4.4, we mentioned that when fitting the Richards function to experimental data, it would be better to solve the problem using the 2 × 2 systems (see section 4.4) rather than by using the 4 × 4 systems (see section 4.3). In fact, we also applied PDS and PFD to the 4 × 4 system using the same starting values, and the results so obtained are summarized in Table 5.7. The advantages of reducing the dimensions (from 4 × 4 to 2 × 2) of this problem are obvious.

TABLE 5.7.: Results obtained by solving the 4 × 4 systems.

Data set	Algorithm	F^* (Sum of squares)	β^*	K^*	n_t	Computing time (seconds)
2" × 2"	PFD	0.5970	30.4617	-59.2143	497	20.41
	PDS	0.6065	17.1710	-33.1656	366	26.49
4" × 4"	PFD	0.4166	9.4032	-18.2857	321	15.50
	PDS	0.4166	9.3999	-18.2797	298	24.97
6" × 6"	PFD	0.6087	10.4301	-17.3652	171	10.36
	PDS	0.6085	10.8927	-18.1391	285	23.94
12" × 12"	PFD	0.6448	11.5092	-20.7368	252	12.86
	PDS	0.6448	11.5581	-20.7908	285	22.38

Our main objective is to compare the performances of the six algorithms (see section 5.3) when they are applied to the problem of solving the 2 × 2 system. To do this, the results given in Tables 5.3-5.6 are summarized in Table 5.8. The ranks given by the Rank I and Rank II columns in each table are added together, and the algorithms are ranked in increasing order of these totals. This number indicates the appro-

priateness of these algorithms for these four problems. Consequently, the algorithms appear in the list according to the ranking of their performances, from best to worst. Moreover, the number of times that each algorithm has failed to solve a problem, or has given an inferior solution are also tabulated in the last two columns in Table 5.8. These last two entries indicate the unreliability of the algorithm.

TABLE 5.8.: Summary of algorithm rankings.

		Data from Tables 5.3-5.6		
Algorithm	Overall Ranking	Sum of ranks	No. of times that produced inferior solution	No. of times that failed to converge
NR	1	13	0	0
NRD	2	15	0	0
PFD	3	20	0	0
PDS	4	32	0	0
ML2	5	43	3	0
PGL2	6	45	0	1

It is clear that, from our experimental results which have been summarized in Table 5.8, NR has shown superiority over the other five techniques. The performance of NRD is close to that of NR and, in fact, when a given set of data is very irregular, NRD seems to be more efficient than NR (see Table 5.3). The efficiency of these two methods is based on the fact that they make use of the second partial derivatives explicitly, and that they have been provided with good starting values.† As might be expected, the performance of PFD far surpassed that of the rest of algorithms, i.e. PDS, PGL2 and ML2. In PFD, the Hessian matrix and its inverse are approximated by some revised formulas and these approximations

†Notice also that we provide special codes for these algorithms based on the formulas on p. 69.

are corrected at each third iteration (see section 3.6). From Tables 5.4 and 5.6, we note that the results obtained by PFD are slightly different from the best results. However, from Table 5.8, we can say that the differences in the performances of NR, NRD and PFD are small.

It is perhaps surprising that PDS appears to be more efficient than PGL2 and ML2. As a direct search method, it is worth noting that PDS has always produced acceptable results to our problems. We pointed out in section 2.5 that the weakness of PGL2 is due to the fact that the second order terms have been truncated in the Taylor series expansion of the objective function. As have been shown, this algorithm failed to converge for this first set of data and it required a large amount of computing time to solve each of the rest of the problems. From Tables 5.2-5.6, we note that ML2 failed to improve the results obtained initially for all sets of data. The fact that this algorithm required a large number of function evaluations has been pointed out in section 3.5. Consequently, PGL2 and ML2 scored the highest rankings and they are not efficient in solving our problems.

Since the present assessment made of these six algorithms is based only on their performances with the Richards function applied to the four data sets of Table 5.1, our findings should not be interpreted too generally. For this particular curve-fitting problem, however, there is little doubt that NR applied to the 2 × 2 system yields satisfactory results.

REFERENCES

(1) BARRODALE, I. (1968). L_1 Approximation and the Analysis of Data. Applied Stat., Vol. 17, pp. 51-57.

(2) BARRODALE, I., and YOUNG, A. (1966). Algorithms for Best L_1 and L_∞ Linear Approximations on a Discrete Set. Numer. Math., Vol. 8, pp. 295-306.

(3) BARRODALE, I., ROBERTS, F.D.K., and HUNT, C.R. (1970). Computing Best L_p Approximations by Functions Nonlinear in One Parameter. The Computer Journal, Vol. 13, #4, pp. 382-386.

(4) BOX, M.J., DAVIS, D., and SWANN, W.H. (1969). Nonlinear Optimization Techniques. I.C.I. Monograph No. 5, Oliver and Boyd Ltd., Edinburgh and London.

(5) CAUSTON, D.R. (1969). A Computer Program for Fitting the Richards Function. Biometrics, Vol. 25, #2, pp. 401-409.

(6) MARQUARDT, D.W. (1963). An Algorithm for Least Squares Estimation of Nonlinear Parameters. SIAM Jour., Vol. 11, pp. 431-441.

(7) NELDER, J.A. (1961). The Fitting of a Generalization of the Logistic Curve. Biometrics 17, pp. 89-110.

(8) POWELL, M.J.D. (1964). An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives. The Computer Journal, Vol. 7, pp. 155-162.

(9) POWELL, M.J.D. (1965). A Method for Minimizing a Sum of Squares of Nonlinear Functions without Calculating Derivatives. The Computer Journal, Vol. 7, pp. 303-307.

(10) POWELL, M.J.D. (1970). A New Algorithm for Unconstrained Optimization. Report No. T.P. 393, A.E.R.E., Harwell,

(11) POWELL, M.J.D. (1970). A FORTRAN Subroutine for Unconstrained Minimization Requiring First Derivatives of the Objective Function, Report No. R.6469, A.E.R.E., Harwell.

(12) POWELL, M.J.D. (1970). Recent Advances in Unconstrained Optimization, Report No. T.P. 430, A.E.R.E., Harwell.

(13) RICE, J.R. (1964). The Approximation of Functions, Vol. 1, Reading, Mass.: Addison-Wesley.

(14) RICE, J.R. (1969). The Approximation of Functions, Vol. 2, Reading, Mass.: Addison-Wesley.

(15) RICHARDS, F.J. (1959). A Flexible Growth Function for Empirical Use. Jour. Exp. Bot., Vol. 10, pp. 290-300.

(16) van den DRIESSCHE, R. (1971). Growth of One-year Old Douglas Fir Plants at Four Spacings. Annals of Botany, to appear.

(17) ZANGWILL, W.I. (1967). Minimizing a Function without Calculating Derivatives. The Computer Journal, Vol. 10, pp. 293-296.

APPENDIX I

THE EXPRESSIONS FOR a AND m

Consider the function defined by (4.9), i.e.

$$F(a, m, \beta, K) = \sum_{i=1}^{N} \{w_i - a - m\ell n(1 + e^{\beta+Kt_i})\}^2 \qquad (A.1)$$

The so-called normal equations, which have to be satisfied at the minimum of (A.1), are:

$$\frac{\partial F}{\partial a} \equiv (-2) \sum_{i=1}^{N} \gamma_i(a, m, \beta, K) = 0 \qquad (A.2)$$

$$\frac{\partial F}{\partial m} \equiv (-2) \sum_{i=1}^{N} \gamma_i(a, m, \beta, K)\, \ell n(1 + e^{\beta+Kt_i}) = 0 \qquad (A.3)$$

$$\frac{\partial F}{\partial \beta} \equiv (-2m) \sum_{i=1}^{N} \{\gamma_i(a, m, \beta, K)e^{\beta+Kt_i}\}/\{1 + e^{\beta+Kt_i}\} = 0 \qquad (A.4)$$

$$\frac{\partial F}{\partial K} \equiv (-2m) \sum_{i=1}^{N} \{\gamma_i(a, m, \beta, K)e^{\beta+Kt_i} t_i\}/\{1 + e^{\beta+Kt_i}\} = 0 \qquad (A.5)$$

where $$\gamma_i(a, m, \beta, K) = w_i - a - m\ell n(1 + e^{\beta+Kt_i}) \qquad (A.6)$$

Solve a and m in terms of β and K from equations (A.2) and (A.3), and we get:

$$a \equiv g_1(\beta, K) = \frac{P_1(\beta, K)}{Q_1(\beta, K)} \qquad (A.7)$$

$$m \equiv g_2(\beta, K) = \frac{P_2(\beta, K)}{Q_2(\beta, K)} \qquad (A.8)$$

where

$$P_1(\beta, K) = (\Sigma w_i)\left[\Sigma u_i^2(\beta, K)\right] - \left[\Sigma u_i(\beta, K)\right]\left[\Sigma w_i u_i(\beta, K)\right] \qquad (A.9)$$

$$Q_1(\beta, K) = N\left[\Sigma u_i^2(\beta, K)\right] - \left[\Sigma u_i(\beta, K)\right]^2 \qquad (A.10)$$

$$P_2(\beta, K) = (\Sigma w_i) - N\, g_1(\beta, K) \qquad (A.11)$$

$$Q_2(\beta, K) = \Sigma u_i(\beta, K) \qquad (A.12)$$

where $u_i(\beta, K) = \ell n(1 + e^{\beta + Kt_i})$. (A.13)

Notice that the symbol Σ represents $\sum_{i=1}^{N}$ for convenience.

Consequently, (4.10) reduces to the form:

$$F(\beta, K) = \sum_{i=1}^{N} s_i^2(\beta, K) \ , \qquad (A.14)$$

where $s_i(\beta, K) = w_i - g_1(\beta, K) - g_2(\beta, K)\, u_i(\beta, K)$. (A.15)

Note that equation (A.14) is equivalent to (4.8), but in different notations. It is evident that the function defined by (A.14) involves two nonlinear parameters β and K only.

APPENDIX II

ELEMENTS OF THE GRADIENT VECTOR AND THE HESSIAN MATRIX

We continue the discussion of Appendix I. Elements of the gradient vector and the Hessian matrix of the function defined in (A.14) will be given in this appendix.

By differentiating $u_i(\beta, K)$ defined in (A.13) successively with respect to β and K, we get:

$$\frac{\partial u_i}{\partial \beta} = e^{\beta+Kt_i}/(1 + e^{\beta+Kt_i})$$

$$\frac{\partial u_i}{\partial K} = t_i \frac{\partial u_i}{\partial \beta}$$

$$\frac{\partial^2 u_i}{\partial \beta^2} = \frac{\partial u_i}{\partial \beta} /(1 + e^{\beta+Kt_i})$$

$$\frac{\partial^2 u_i}{\partial \beta \partial K} = t_i \frac{\partial^2 u_i}{\partial \beta^2} = \frac{\partial^2 u_i}{\partial K \partial \beta}$$

$$\frac{\partial^2 u_i}{\partial K^2} = t_i \frac{\partial^2 u_i}{\partial \beta \partial K}$$

Consequently, from (A.9)-(A.12), we get:

$$\frac{\partial P_1}{\partial \beta} = 2(\Sigma w_i)(\Sigma u_i \frac{\partial u_i}{\partial \beta}) - (\Sigma \frac{\partial u_i}{\partial \beta})(\Sigma w_i u_i) - (\Sigma u_i)(\Sigma w_i \frac{\partial u_i}{\partial \beta})$$

$$\frac{\partial P_1}{\partial K} = 2(\Sigma w_i)(\Sigma u_i \frac{\partial u_i}{\partial K}) - (\Sigma \frac{\partial u_i}{\partial K})(\Sigma w_i u_i) - (\Sigma u_i)(\Sigma w_i \frac{\partial u_i}{\partial K})$$

$$\frac{\partial Q_1}{\partial \beta} = 2\left[N(\Sigma u_i \frac{\partial u_i}{\partial \beta}) - (\Sigma u_i)(\Sigma \frac{\partial u_i}{\partial \beta})\right]$$

$$\frac{\partial Q_1}{\partial K} = 2\left[N(\Sigma u_i \frac{\partial u_i}{\partial K}) - (\Sigma u_i)(\Sigma \frac{\partial u_i}{\partial K})\right]$$

It follows from (A.7) and (A.8) that:

$$\frac{\partial g_1}{\partial \beta} = \left(\frac{\partial P_1}{\partial \beta} - g_1 \frac{\partial Q_1}{\partial \beta}\right)/Q_1$$

$$\frac{\partial g_1}{\partial K} = \left(\frac{\partial P_1}{\partial K} - g_1 \frac{\partial Q_1}{\partial K}\right)/Q_1$$

$$\frac{\partial g_2}{\partial \beta} = \left[-N \frac{\partial g_1}{\partial \beta} - g_2(\Sigma \frac{\partial u_i}{\partial \beta})\right]/Q_2$$

$$\frac{\partial g_2}{\partial K} = \left[-N \frac{\partial g_1}{\partial K} - g_2(\Sigma \frac{\partial u_i}{\partial K})\right]/Q_2$$

Since $s_i(\beta, K)$ is defined by (A.15), we have:

$$\frac{\partial s_i}{\partial \beta} = -\frac{\partial g_1}{\partial \beta} - \frac{\partial g_2}{\partial \beta} u_i - g_2 \frac{\partial u_i}{\partial \beta}$$

$$\frac{\partial s_i}{\partial K} = -\frac{\partial g_1}{\partial K} - \frac{\partial g_2}{\partial K} u_i - g_2 \frac{\partial u_i}{\partial K}$$

From equations (A.4) and (A.5), elements of the gradient vector are given by:

$$\frac{\partial F}{\partial \beta} \equiv F_1(\beta, K) = \Sigma \{s_i(\beta, K) \frac{\partial u_i}{\partial \beta}\}$$

$$\frac{\partial F}{\partial K} \equiv F_2(\beta, K) = \Sigma \{s_i(\beta, K) \frac{\partial u_i}{\partial K}\}$$

Elements of the Hessian matrix of (4.11) are as follows:

$$\frac{\partial^2 F}{\partial \beta^2} \equiv F_{11}(\beta, K) = \Sigma\left[\frac{\partial s_i}{\partial \beta}\frac{\partial u_i}{\partial \beta} + s_i \frac{\partial^2 u_i}{\partial \beta^2}\right]$$

$$\frac{\partial^2 F}{\partial \beta \partial K} \equiv F_{12}(\beta, K) = \Sigma\left[\frac{\partial s_i}{\partial \beta}\frac{\partial u_i}{\partial K} + s_i \frac{\partial^2 u_i}{\partial \beta \partial K}\right]$$

$$\frac{\partial^2 F}{\partial K \partial \beta} \quad F_{21}(\beta, K) = \frac{\partial^2 F}{\partial \beta \partial K}$$

$$\frac{\partial^2 F}{\partial K^2} \quad F_{22}(\beta, K) = \Sigma\left[\frac{\partial s_i}{\partial K}\frac{\partial u_i}{\partial K} + s_i \frac{\partial^2 u_i}{\partial K^2}\right]$$

The iterative formula of the Newton-Raphson method in this 2-dimensional case at the r^{th} stage is then given by:

$$\beta^{(r+1)} = (F_2 F_{12} - F_1 F_{22})/D + \beta^{(r)}$$

$$K^{(r+1)} = (F_1 F_{21} - F_2 F_{11})/D + K^{(r)}$$

where $D = (F_{11}F_{22} - F_{12}F_{21})$ and all the derivatives are evaluated at the point $(\beta^{(r)}, K^{(r)})$.

MAIN PROGRAM FOR THE NEWTON-RAPHSON METHOD

```
      IMPLICIT REAL *8 (A-H,O-Z)
C
C     THIS MAIN PROGRAM IS TO ILLUSTRATE HOW TO LINK  THOSE SUBROUTINES
C     TOGETHER
C
C     N  =NUMBER OF DATA POINTS
C     B  =THE VECTOR CONTAINING BETA AND K IN EQUATION (4.8)
C     F  =THE SUM OF SQUARES DEFINED IN EQUATION (4.8)
C     MAX=MAXIMUM NUMBER OF ITERATIONS ALLOWED
C     EPS=THE REQUIRED ACCURACY
C     P  =THE GRADIENT VECTOR GIVEN IN APPENDIX II
C     G  =THE HESSIAN MATRIX GIVEN IN APPENDIX II
C     W  =THE VECTOR CONTAINING THE DATA OF WEIGHTS
C     T  =THE VECTOR CONTAINING THE DATA OF TIMES
C     Q  =THE VECTOR CONTAINING A AND M GIVEN IN APPENDIX I
C
      COMMON /BLK 1/ W(21),T(21),Q(2)
      COMMON /BLK 2 / P(2),G(2,2)
      DIMENSION B(2)
      WRITE(6,100)
  100 FORMAT('1')
C
C     INPUT THE DATA
C
      N=21
      MAX=10
      EPS=1.0D-03
      DO 1 I=1,20
    1 T(I)=(I-1)*14.0/365.0
      T(21)=21.0*14.0/365.0
      READ (5,103)(W(I),I=1,N)
  103 FORMAT(16F5.3)
      WRITE(6,104)(T(I),I=1,N)
      WRITE(6,104)(W(I),I=1,N)
  104 FORMAT('0',7D18.8)
```

```
      READ(5,105)(B(I),I=1,2)
  105 FORMAT(4F9.5)
C
C     CALCULATE THE STARTING VALUES
C
      CALL START(N,B)
      DO 2 I=1,N
    2 W(I)=DLOG(W(I))
C
C     SOLVE THE 2*2 BY THE NEWTON-RAPHSON METHOD
C
      CALL NEWTON(N,B,F,MAX,EPS)
      CALL EXIT
      END
```

SUBROUTINE FOR CALCULATING THE STARTING VALUES

```
      SUBROUTINE START(N,B)
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /BLK 1/ W(21),T(21),Q(2)
      DIMENSION R(21),WC(21),B(2)
C
C     THIS SUBROUTINE IS TO CALCULATE THE STARTING VALUES OF THE
C     PARAMETERS IN EQUATION (4.8) BASED ON THE METHOD SUGGESTED BY
C     CAUSTON, C. R. (1969), BIOMETRICS, VOL.25, P.401-409.
C     N  =NUMBER OF DATA POINTS
C     B  =THE VECTOR CONTAINING BETA AND K IN EQUATION (4.8)
C     W  =THE VECTOR CONTAINING THE DATA OF WEIGHTS
C     T  =THE VECTOR CONTAINING THE DATA OF TIMES
C     Q  =THE VECTOR CONTAINING A AND M GIVEN IN APPENDIX I
C
      IC=0
      SR=0.D0
      C=1.0D0
      IN=N-2
C
C     COMPUTE R(J)'S
C
      DO 1 I=1,IN
      R(I+1)=(DLOG(W(I+2))-DLOG(W(I)))/(T(I+2)-T(I))
      SR=SR+R(I+1)
    1 CONTINUE
      RM=SR/IN
C
C     EXECUTE LINEAR REGRESSION
C
    2 SW=0.D0
      RW=0.D0
      WW=0.D0
      DO 21 I=1,IN
      WC(I+1)=W(I+1)**C
      SW=SW+WC(I+1)
```

```
   21 CONTINUE
      WM=SW/IN
      DO 22 I=1,IN
      RW=RW+(R(I+1)-RM)*(WC(I+1)-WM)
      WW=WW+(WC(I+1)-WM)**2
   22 CONTINUE
      BETA=RW/WW
      ALPHA=RM-BETA*WM
C
C     DETERMINE THE OPTIMAL VALUE OF N
C
      RES=0.D0
      DO 3 I=1,IN
      RES=RES+(R(I+1)-ALPHA-BETA*WC(I+1))**2
    3 CONTINUE
      IC=IC+1
      IF(IC.EQ.1)GO TO 31
      IF(DABS(C).LT.1.D-10)GO TO 33
      IF(RES.GT.RESI)GO TO 32
   31 RESI=RES
   33 C=C+0.1D0
      GO TO 2
   32 CONTINUE
C
C     COMPUTE STARTING VALUES OF THE REST OF PARAMETERS
C
      SY=0.D0
      ST=0.D0
      M=0
      CK=ALPHA*C
      IF(CK.LT.0.D0)GO TO 99
      A=(-(CK/(BETA*C)))**(1.D0/C)
      IN=N-1
      DO 4 I=2,IN
      IF(W(I).GE.A)GO TO 4
```

```
      M=M+1
   41 SY=SY+DLOG(DABS((A/W(I))**C-1.D0))
      ST=ST+T(I)
    4 CONTINUE
      B(1)=(SY+CK*ST)/M
      B(2)=-CK
      Q(1)=DLOG(A)
      Q(2)=(-1.0)/C
      IF(C.GT.0.D0)GO TO 100
      B(2)=-B(2)
      GO TO 100
   99 WRITE(6,10)
   10 FORMAT('-','THE VALUE OF K IS NEGATIVE ')
  100 RETURN
      END
```

SUBROUTINE FOR THE NEWTON-RAPHSON METHOD

```
      SUBROUTINE NEWTON (N,B,F,MAX,EPS)
C
C     THIS SUBROUTINE IS TO SOLVE THE 2*2 SYSTEM (SEE SECTION 4.4) BY
C     USING THE NEWTON-RAPHSON METHOD.
C
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /BLK 1/ W(21),T(21),Q(2)
      COMMON /BLK 2 / P(2),G(2,2)
      DIMENSION B(2)
C
C     N   =NUMBER OF DATA POINTS
C     B   =THE VECTOR CONTAINING BETA AND K IN EQUATION (4.8)
C     F   =THE SUM OF SQUARES DEFINED IN EQUATION (4.8)
C     MAX=MAXIMUM NUMBER OF ITERATIONS ALLOWED
C     EPS=THE REQUIRED ACCURACY
C     P   =THE GRADIENT VECTOR GIVEN IN APPENDIX II
C     G   =THE HESSIAN MATRIX GIVEN IN APPENDIX II
C     W   =THE VECTOR CONTAINING THE DATA OF WEIGHTS
C     T   =THE VECTOR CONTAINING THE DATA OF TIMES
C     Q   =THE VECTOR CONTAINING A AND M GIVEN IN APPENDIX I
C
      IC=0
C
C     BEGIN AN ITERATION BY TESTING FOR CONVERGENCE
C
   11 IC=IC+1
      IF (IC.GE.MAX)GO TO 13
      CALL VALUE (N,B,F)
      WRITE(6,100)IC,F
  100 FORMAT('-','AFTER',I3,'  ITERATIONS',5X,'FUNCTION VALUE =',D20.10)
      WRITE(6,101)(B(I),I=1,2)
  101 FORMAT('0','X=',2D20.10)
      WRITE(6,102)(P(I),I=1,2)
  102 FORMAT('0','G =',2D20.10)
      IF (DABS(P(1)).LE.EPS.AND.DABS(P(2)).LE.EPS)GO TO 12
```

```
      D=G(1,1)*G(2,2)-G(1,2)*G(2,1)
      D1=(G(1,2)*P(2)-G(2,2)*P(1))/D
      D2=(G(2,1)*P(1)-G(1,1)*P(2))/D
      B(1)=B(1)+D1
      B(2)=B(2)+D2
      GO TO 11
C
C     OUTPUT THE FINAL RESULTS
C
   12 WRITE(6,103)
  103 FORMAT('-','THE FINAL RESULTS ARE AS FOLLOWS :')
      WRITE(6,100)IC,F
      WRITE(6,101)(B(I),I=1,2)
      WRITE(6,104)(Q(I),I=1,2)
  104 FORMAT('0','VALUES OF A AND M ARE :',2D20.10)
      GO TO 14
   13 WRITE(6,105)IC
  105 FORMAT('-','THE NUMBER OF ITERATIONS HAS EXCEEDED ',I3)
      WRITE(6,106)
  106 FORMAT('0','THE RESULTS OBTAINED AT THIS STAGE ARE AS FOLLOWS :')
      WRITE(6,107)F
  107 FORMAT('-','F =',D20.10)
      WRITE(6,101)(B(I),I=1,2)
      WRITE(6,104)(Q(I),I=1,2)
   14 RETURN
      END
```

SUBROUTINE FOR THE NEWTON-RAPHSON METHOD WITH LINEAR SEARCH

```
      SUBROUTINE NRDAVI (N,B,F,MAX,EPS)
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /BLK 1/ W(21),T(21),Q(2)
      COMMON /BLK 2 / P(2),G(2,2)
      DIMENSION B(2)
C
C     THIS SUBROUTINE IS TO SOLVE THE 2*2 SYSTEM BY USING THE NEWTON-
C     RAPHSON METHOD WITH DAVIDON'S LINEAR SEARCH TECHNIQUE.
C
C     N  =NUMBER OF DATA POINTS
C     B  =THE VECTOR CONTAINING BETA AND K IN EQUATION (4.8)
C     F  =THE SUM OF SQUARES DEFINED IN EQUATION (4.8)
C     MAX=MAXIMUM NUMBER OF ITERATIONS ALLOWED
C     EPS=THE REQUIRED ACCURACY
C     P  =THE GRADIENT VECTOR GIVEN IN APPENDIX II
C     G  =THE HESSIAN MATRIX GIVEN IN APPENDIX II
C     W  =THE VECTOR CONTAINING THE DATA OF WEIGHTS
C     T  =THE VECTOR CONTAINING THE DATA OF TIMES
C     Q  =THE VECTOR CONTAINING A AND M GIVEN IN APPENDIX I
C
      IC=0
C
C     BEGIN AN ITERATION BY TESTING FOR CONVERGENCE
C
   11 IC=IC+1
      IF (IC.GE.MAX)GO TO 13
   15 CALL VALUE (N,B,F)
      WRITE(6,100)IC,F
  100 FORMAT('-','AFTER',I3,'  ITERATIONS',5X,'FUNCTION VALUE =',D20.10)
      WRITE(6,101)(B(I),I=1,2)
  101 FORMAT('0','X=',2D20.10)
      WRITE(6,102)(P(I),I=1,2)
  102 FORMAT('0','G =',2D20.10)
      IF (DABS(P(1)).LE.EPS.AND.DABS(P(2)).LE.EPS)GO TO 12
C
```

```
C       FIND THE VALUE OF LAMBDA 1
C
        D=G(1,1)*G(2,2)-G(1,2)*G(2,1)
        D1=(G(1,2)*P(2)-G(2,2)*P(1))/D
        D2=(G(2,1)*P(1)-G(1,1)*P(2))/D
        XLAM=2.0
        U=P(1)*D1+P(2)*D2
        FS=0.4
        Z=((-2.0)*(F-FS))/U
        IF (Z.LT.2.0)XLAM=Z
C
C       TEST IF THE MINIMUM HAS BEEN STRADDLED
C
        B(1)=B(1)+XLAM*D1
        B(2)=B(2)+XLAM*D2
        CALL VALUE (N,B,Y)
        V=P(1)*D1+P(2)*D2
        IF (Y.GE.F.OR.V.GE.0.0)GO TO 20
        GO TO 15
C
C       A CUBIC IS THEN FITTED THROUGTH THESE TWO POINTS
C
   20   Z=(3.0*(F-Y))/XLAM+U+V
        VV=DSQRT(DABS(Z**2-U*V))
        UV=(V+VV-Z)/(V-U+2.0*VV)
        YLAM=XLAM*(1.0-UV)
        B(1)=B(1)+(YLAM-XLAM)*D1
        B(2)=B(2)+(YLAM-XLAM)*D2
        GO TO 11
C
C       OUTPUT THE FINAL RESULTS
C
   12   WRITE(6,103)
  103   FORMAT('-','THE FINAL RESULTS ARE AS FOLLOWS :')
        WRITE(6,100)IC,F
```

```
      WRITE(6,101)(B(I),I=1,2)
      WRITE(6,104)(Q(I),I=1,2)
  104 FORMAT('0','VALUES OF A AND M ARE :',2D20.10)
      GO TO 14
   13 WRITE(6,105)IC
  105 FORMAT('-','THE NUMBER OF ITERATIONS HAS EXCEEDED ',I3)
      WRITE(6,106)
  106 FORMAT('0','THE RESULTS OBTAINED AT THIS STAGE ARE AS FOLLOWS :')
      WRITE(6,107)F
  107 FORMAT('-','F =',D20.10)
      WRITE(6,101)(B(I),I=1,2)
      WRITE(6,104)(Q(I),I=1,2)
   14 RETURN
      END
```

SUBROUTINE FOR CALCULATING THE PARTIAL DERIVATIVES

```
      SUBROUTINE VALUE(N,B,F)
C
C     THIS SUBROUTINE IS TO CALCULATE THE FIRST AND SECOND PARTIAL
C     DERIVATIVES GIVEN IN APPENDIX II.
C
C     N  =NUMBER OF DATA POINTS
C     B  =THE VECTOR CONTAINING BETA AND K IN EQUATION (4.8)
C     F  =THE SUM OF SQUARES DEFINED IN EQUATION (4.8)
C     P  =THE GRADIENT VECTOR GIVEN IN APPENDIX II
C     G  =THE HESSIAN MATRIX GIVEN IN APPENDIX II
C     W  =THE VECTOR CONTAINING THE DATA OF WEIGHTS
C     T  =THE VECTOR CONTAINING THE DATA OF TIMES
C     Q  =THE VECTOR CONTAINING A AND M GIVEN IN APPENDIX I
C     S  =WORKING AREA
C
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /BLK 1/ W(21),T(21),Q(2)
      COMMON /BLK 2 / P(2),G(2,2)
      DIMENSION B(2),S(19)
C
C     EXECUTE THE SUMMATIONS
C
      DO 11 I=1,19
      S(I)=0.0D+00
   11 CONTINUE
      DO 1 I=1,N
      E=DEXP(B(1)+B(2)*T(I))
      EE=1.0D+00+E
      H=DLOG(EE)
      DHB=E/EE
      DHK=DHB*T(I)
      DDHB=DHB/EE
      DHBK=DDHB*T(I)
      DDHK=DHBK*T(I)
      S(1)=S(1)+W(I)
```

```
      S(2)=S(2)+H
      S(3)=S(3)+H*W(I)
      S(4)=S(4)+H**2
      S(5)=S(5)+DHB
      S(6)=S(6)+DHK
      S(7)=S(7)+DHB*W(I)
      S(8)=S(8)+DHK*W(I)
      S(9)=S(9)+DHB*H
      S(10)=S(10)+DHK*H
      S(11)=S(11)+DHB**2+H*DDHB
      S(12)=S(12)+DDHB
      S(13)=S(13)+DDHB*W(I)
      S(14)=S(14)+DHB*DHK+H*DHBK
      S(15)=S(15)+DHBK
      S(16)=S(16)+DHBK*W(I)
      S(17)=S(17)+DDHK**2+H*DDHK
      S(18)=S(18)+DDHK
      S(19)=S(19)+DDHK*W(I)
C
C     CALCULATE THE PARTIAL DERIVATIVES OF A AND M
C
    1 CONTINUE
      P1=S(1)*S(4)-S(2)*S(3)
      Q1=N*S(4)-S(2)**2
      Q(1)=P1/Q1
      Q(2)=(S(1)-N*Q(1))/S(2)
      P11=(2.0D+00)*S(1)*S(9)-S(5)*S(3)-S(2)*S(7)
      P12=(2.0D+00)*S(1)*S(10)-S(6)*S(3)-S(2)*S(8)
      Q11=(2.0D+00)*(N*S(9)-S(2)*S(5))
      Q12=(2.0D+00)*(N*S(10)-S(2)*S(6))
      DP11=2.0*S(1)*S(11)-S(3)*S(12)-2.0*S(5)*S(7)-S(2)*S(16)
      DP12=2.0*S(1)*S(14)-S(3)*S(15)-S(5)*S(8)-S(6)*S(7)-S(2)*S(16)
      DP22=2.0*S(1)*S(17)-S(3)*S(18)-2.0*S(6)*S(8)-S(2)*S(19)
      DQ11=(2.0)*(N*S(11)-S(5)**2-S(2)*S(12))
      DQ12=(2.0)*(N*S(14)-S(5)*S(6)-S(2)*S(15))
```

```
      DQ22=(2.0)*(N*S(17)-S(6)**2-S(2)*S(18))
      G11=(P11-Q(1)*Q11)/Q1
      G12=(P12-Q(1)*Q12)/Q1
      DG11=(DP11-Q(1)*DQ11-((2.0)*(P11-Q(1)*Q11)*Q11)/Q1)/Q1
      DG12=(DP12-Q(1)*DQ12-(Q12*(P11-Q(1)*Q11)+Q11*(P12-Q(1)*Q12))/Q1)/
     /Q1
      DG22=(DP22-Q(1)*DQ22-((2.0)*(P12-Q(1)*Q12)*Q12)/Q1)/Q1
      P1=S(1)-N*Q(1)
      Q1=S(2)
      P11=-N*G11
      P12=-N*G12
      Q11=S(5)
      Q12=S(6)
      DP11=-N*DG11
      DP12=-N*DG12
      DP22=-N*DG22
      DQ11=S(12)
      DQ12=S(15)
      DQ22=S(18)
      DMB=(P11-Q(2)*Q11)/Q1
      DMK=(P12-Q(2)*Q12)/Q1
      DMBB=(DP11-Q(2)*DQ11-((2.0)*(P11-Q(2)*Q11)*Q11)/Q1)/Q1
      DMBK=(DP12-Q(2)*DQ12-(Q12*(P11-Q(2)*Q11)+Q11*(P12-Q(2)*Q12))/Q1)/
     /Q1
      DMKK=(DP22-Q(2)*DQ22-((2.0)*(P12-Q(2)*Q12)*Q12)/Q1)/Q1
C
C     EVALUATE THE GRADIENT VECTOR AND THE HESSIAN MATRIX
C
      F=0.0D+00
      P(1)=0.0D+00
      P(2)=0.0D+00
      G(1,1)=0.0D+00
      G(1,2)=0.0D+00
      G(2,2)=0.0D+00
      DO 2 I=1,N
```

```
      E=DEXP(B(1)+B(2)*T(I))
      EE=1.0D+00+E
      H=DLOG(EE)
      DHB=E/EE
      DHK=DHB*T(I)
      DDHB=DHB/EE
      DHBK=DDHB*T(I)
      DDHK=DHBK*T(I)
      R=W(I)-Q(1)-Q(2)*H
      F=F+R**2
      U=G11+DMB*H+Q(2)*DHB
      V=G12+DMK*H+Q(2)*DHK
      P(1)=P(1)+R*U
      P(2)=P(2)+R*V
      G(1,1)=G(1,1)-U**2+R*(DG11+DMBB*H+2.0*DMB*DHB+Q(2)*DDHB)
      G(2,2)=G(2,2)-V**2+R*(DG22+DMKK*H+(2.0)*DMK*DHK+Q(2)*DDHK)
      G(1,2)=G(1,2)-U*V+R*(DG12+DMBK*H+DMB*DHK+DMK*DHB+Q(2)*DHBK)
    2 CONTINUE
      G(2,1)=G(1,2)
      DO 3 I=1,2
      P(I)=(-2.0D+00)*P(I)
      DO 3 J=1,2
      G(I,J)=(-2.0D+00)*G(I,J)
    3 CONTINUE
      RETURN
      END
```

VITA

Surname: PHUA Given Names: KANG-HOH

Place of Birth: SINGAPORE Date of Birth: JUNE 20, 1943

Educational Institutions Attended, with Dates of Entering and Leaving:

NANYANG UNIVERSITY (SINGAPORE)	1965 to 1969
	____ to ____
	____ to ____
	____ to ____

Degrees, Diplomas, Etc., Awarded, with Dates and Names of Institutions:

B.Sc.	1969	NANYANG UNIVERSITY

Honors and Awards:

The Singapore Government University Scholarship, 1965/1966

Nanyang University Assistantship, 1967/1968 and 1968/1969

University of Victoria Fellowship, 1969/1970 and 1970/1971

Publications:

THE UNIVERSITY OF VICTORIA LIBRARY

MANUSCRIPT THESIS

AUTHORITY TO DISTRIBUTE

AUTHOR: This dissertation may be lent or microfilm copies made available:

(a) Without restriction [redacted]

(b) With the restriction that, for a period of five years (until _____) the written approval of the following is required:

(1) The Chairman, School of Graduate Studies _____

(2) The Author _____

(3) both the Chairman, School of Graduate Studies, and the Author _____

BORROWERS: The borrower undertakes, by signing below, to give proper credit for any use made of the dissertation, and to obtain the consent of the author if it is proposed to make extensive quotations, or to reproduce the dissertation in whole or in part,

Signature of Borrower	Address	Date