PARAMETER RECLASSIFICATION IN
NONLINEAR LEAST SQUARES

by

Richard Hugh Barham

Technical Report No. 97
Department of Statistics NASA Contract

February 26, 1971

Research sponsored by National Aeronautics and Space Administration
Grant No. NGR 44-007-006

DEPARTMENT OF STATISTICS
Southern Methodist University

PARAMETER RECLASSIFICATION IN
NONLINEAR LEAST SQUARES

A Dissertation Presented to the Faculty of the Graduate School

of

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Statistics

by

Richard Hugh Barham
(B.A., North Texas State University, 1952)

February 26, 1971

Barham, Richard Hugh

B.A., North Texas State
University, 1952

Parameter Reclassification In Nonlinear Least Squares

Adviser: Associate Professor Wanzer Drane

Doctor of Philosophy degree conferred May 23, 1971

Dissertation completed February 26, 1971

A procedure is presented for nonlinear least squares estimation in which the parameters to be estimated are reclassified from all nonlinear to linear-nonlinear. The theoretical basis for the reclassification approach is presented together with a discussion of convergence criteria and confidence regions using this method. Examples are presented which allow a comparison of the nonlinear to the linear-nonlinear method employing two widely used iterative techniques, those of Hartley and Marquardt. The reclassification method reduces the dimensionality of the vector of iterants and thus the number of initial guesses to be made and simplifies the sum-of-squared-error surface. In many cases, this reduction affords faster convergence because of less iterations required. Improved results (less iterations and/or computer run time) are obtained for the linear-nonlinear method when using the Hartley technique, but not when using the Marquardt technique.

ACKNOWLEDGMENTS

The fellowship from Mobil Research and Development Corporation made my graduate studies possible. I will always be grateful for the opportunity afforded me to extend my knowledge, stimulate my thinking, and most of all broaden my associations to include faculty and graduate students interested in and devoted to the field of statistics. I wish to express my deep appreciation to Associate Professor J. Wanzer Drane for suggesting this dissertation topic and for his guidance and assistance throughout my research.

The computational work for this research was supported by NASA multidisciplinary grant to SMU, No. 83-36. Much of the algorithm was written by B. A. Blumenstein (Emory University) and was modified by Dr. W. R. Schucany (Manager, SMU Statistics Laboratory) in 1968. Dr. Schucany's assistance along with that of W. H. Frawley from the Statistics Laboratory in the revision of that routine for this research is greatly appreciated and acknowledged. I wish to acknowledge the faculty of the Department of Statistics at Southern Methodist University for their courses and assistance during my enrollment at the University. All offered encouragement to me at times when it was very much needed.

My wife Jean and daughters Julie, Jeanie, and Jill gave much love, understanding, and encouragement to me during my graduate studies. I could not have completed my work without their support. I want to thank Mrs. Nancy Mitchell for her excellent job of typing this dissertation.

TABLE OF CONTENTS

Page

Chapter

Appendix

LIST OF TABLES

CHAPTER I

INTRODUCTION

In science and industry data are continually being gathered as the result of experiments made to further man's search for knowledge to better his life. There is a need, then, to have techniques available to extract from these data hidden or implied relationships. It is desirable to be able to approximate the functional relationship between certain response observations and corresponding inputs by a mathematical model. Many models which describe best the desired functional relationship contain nonlinear as well as linear parameters. A nonlinear regression problem is one in which at least one of the parameters to be estimated enters the model in a nonlinear manner.

Consider the problem of fitting a nonlinear regression model to a set of n observations of the form y_h, x_{1h}, x_{2h}, ..., x_{kh}, where y_h is the h^{th} response, h = 1, 2, ..., n to a set of inputs x_{1h}, x_{2h}, ..., x_{kh}. The model to which this set of observations is to be fitted can be written in the form

$$y_h = f(x_{1h}, x_{2h}, \ldots, x_{kh}; \Theta_1, \Theta_2, \ldots, \Theta_m) + e_h \qquad (1)$$

where Θ_1, Θ_2, ..., Θ_m are the parameters to be estimated and the errors e_h arise from independent normal distributions; i.e., $\underline{e} = N_n(\underline{0}, \sigma^2 I)$ where $\underline{e} = (e_1, e_2, \ldots, e_n)'$, $\underline{0}$ is the n × 1 zero vector and I the n × n identity matrix.

The model given by equation (1) can be written in vector notation as

$$E(y_h) = f(\underline{x}_h;\underline{\theta}) \tag{2}$$

where

$$\underline{x}_h = (x_{1h}, x_{2h}, \ldots, x_{kh})'$$

$$\underline{\theta} = (\theta_1, \theta_2, \ldots, \theta_m)'$$

The method most frequently employed for obtaining the estimate of the unknown parameter $\underline{\theta}$ is the method of least squares. A set of $\theta_i (i=1,2,\ldots,m)$ for which the error sum of squares

$$Q(\underline{\theta}) = \sum_{h=1}^{n} [y_h - f(\underline{x}_h;\underline{\theta})]^2 \tag{3}$$

is a minimum must be determined. $Q(\underline{\theta})$ is a function of $\underline{\theta}$ only since y_h and $\underline{x}_h$ are fixed observations. Let the value of $\underline{\theta}$ which minimizes $Q(\underline{\theta})$ be $\hat{\underline{\theta}}$. This least squares estimate of $\underline{\theta}$ is obtained by differentiating equation (3) with respect to θ_i and setting the result equal to zero, yielding

$$\frac{\partial Q(\underline{\theta})}{\partial \theta_i} = -2 \sum_{h=1}^{n} [y_h - f(\underline{x}_h;\underline{\theta})] \frac{\partial f(\underline{x}_h;\underline{\theta})}{\partial \theta_i} = 0, \tag{4}$$

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

This gives m normal equations which must be solved for $\hat{\underline{\theta}}$. When the regression function $f(\underline{x}_h;\underline{\theta})$ is linear in the parameters, the m normal equations are linear equations in θ_i and when $f(\underline{x}_h;\underline{\theta})$ is nonlinear in the θ's, the m normal equations will also be nonlinear.

The solution to equation (4) is in general not obtainable in closed form. Numerical methods are thus used to obtain the least squares estimate

$\hat{\underline{\theta}}$. Most iterative algorithms for the least squares estimation of nonlinear parameters use a modified Gauss-Newton method, steepest descent, or a combination of these two methods. Two of the most widely used algorithms are ones utilizing a modified Gauss-Newton method due to Hartley[7] and a combination of the modified Gauss-Newton method with the method of steepest descent due to Marquardt [11]. These are iterative procedures and must begin with a starting (initial) guess, say $\underline{\theta}^0$, for the entire set of parameters $\underline{\theta}$. This initial estimate is improved in subsequent iterations by calculating a correction factor to each of the nonlinear parameters until the correction factor and/or the difference in the error sum of squares at each iteration becomes sufficiently small. The sequence $\{\underline{\theta}^i\}$ converges to $\underline{\theta}^{min}(\hat{\underline{\theta}})$ under conditions discussed in references [7] and [11]. Convergence criteria will be discussed more fully in Chapter III.

For any given mathematical model the number of iterations necessary for convergence of the sequence $\{\underline{\theta}^i\}$ to $\underline{\theta}^{min}$ depends upon the observations, the starting vector and the algorithm used. Conversely, for any given set of data and algorithm, the model chosen for fitting these data determines the rate of convergence and, in some instances, whether the iterative procedure will converge at all. One aspect of this subject has been discussed by comparing results for particular models and data using five different algorithms (modified Gauss-Newton method) by Flanagan, Vitale, and Mendelsohn [5]. Spang [20] reviewed minimization techniques for nonlinear estimation. Jennrich and Sampson [9] and Marquardt [12,13] developed techniques applicable to situations where conditions necessary for convergence are not satisfied when the aforementioned methods are employed. These methods are reserved for these specialized problems since they are more complicated usually requiring greater computer time. Ross [18] discusses methods which can be used for nonlinear parameter estimation when a particular iterative technique fails.

Drane and Schucany [3] considered another approach to the nonlinear least squares problem. They observed that any parameter in least squares regression can be classified as either linear or nonlinear in a given mathematical model and need not necessarily be analyzed as all linear or all nonlinear (if at least one nonlinear parameter exists in the model). Richards [17] had eluded to this aspect of the problem, but only for a specialized problem.

Equation (1) can thus be written as

$$y_h = \sum_{j=1}^{p} \beta_j g_j(x_{1h}, x_{2h}, \ldots, x_{kh}; \alpha_1, \alpha_2, \ldots, \alpha_q) + e_h \tag{5}$$

where β_1, β_2, ..., β_p represent the linear parameters and $\alpha_1, \alpha_2, \ldots, \alpha_q$ represent the nonlinear parameters in the model. In addition, $g_j(x_{1h}, x_{2h}, \ldots, x_{kh}; \alpha_1, \alpha_2, \ldots, \alpha_q)$ are functions of the nonlinear parameters only and the k input variables, x_{1h}, x_{2h}, ..., x_{kh}. The model still consists of a total of m parameters, but they have been reclassified as p linear and q nonlinear parameters where p + q = m. Equation (5) can be written in matrix notation as

$$E(\underline{y}) = F\underline{\beta} \tag{6}$$

where the expected value of a vector $\underline{y}$ of n observations is equal to a linear combination of the p-columns of the n × p F matrix of linearly independent functions $g_j(\underline{x}_h; \underline{\alpha})$, j = 1, 2, ..., p; h = 1, 2, ..., n; $\underline{x}_h = (x_{1h}, x_{2h}, \ldots, x_{kh})'$; $\underline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_q)'$, $\underline{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)'$, and

$$F = \begin{bmatrix} g_1(\underline{x}_1;\underline{\alpha}) & \cdots & g_p(\underline{x}_1;\underline{\alpha}) \\ g_1(\underline{x}_2;\underline{\alpha}) & \cdots & g_p(\underline{x}_2;\underline{\alpha}) \\ \vdots & & \vdots \\ g_1(\underline{x}_n;\underline{\alpha}) & \cdots & g_p(\underline{x}_n;\underline{\alpha}) \end{bmatrix}$$

In the above, $\underline{\beta}$ is the vector of p linear parameters and $\underline{\alpha}$ the vector of q nonlinear parameters. The error sum of squares in matrix notation then becomes

$$Q(\underline{\theta}) = Q(\underline{\beta},\underline{\alpha}) = (\underline{y}-F\underline{\beta})'(\underline{y}-F\underline{\beta}) \tag{7}$$

The problem then becomes one of calculating p $\hat{\beta}$'s and q $\hat{\alpha}$'s, the least squares estimators of $\underline{\beta}$ and $\underline{\alpha}$, which will yield a minimum $Q(\underline{\beta},\underline{\alpha})$. See Drane [2].

The approach used by Drane and Schucany was to determine for an initial set of q nonlinear parameters, say $\underline{\alpha}^0$, the value of $\underline{\beta}$ which minimizes $Q(\underline{\beta}|\underline{\alpha}^0)$. This can be done by the usual linear regression technique to give

$$\underline{\beta}^0 = (F_0'F_0)^{-1}F_0'\underline{y} \tag{8}$$

They then treated $E(\underline{y})$ directly as a nonlinear problem with $m - p = q$ parameters and employed the Hartley or Marquardt iterative procedure to calculate an improved value of $\underline{\alpha}^0$, say $\underline{\alpha}^1$. Obtainment of $\underline{\alpha}^1$ allowed $\underline{\beta}^1$ to be calculated by equation (8). This procedure was repeated until $\underline{\alpha}^i$ converged to $\underline{\alpha}^{min}(\hat{\underline{\alpha}})$ wherein $\underline{\beta}^{min}(\hat{\underline{\beta}})$ could be calculated. Results indicated this procedure not to be any better (as far as computer time or iterations to convergence) than the technique of considering the β as well as the α parameters as nonlinear and using the Hartley or Marquardt techniques directly. Some specific results are discussed in reference [3] from which comparisons can be made to results presented in Chapter IV using the algorithm to be presented.

Walling [23] used essentially the same approach as Drane and Schucany, but considered the inner-relationship between the linear and nonlinear parameters when using the Gauss-Newton iterative algorithm to calculate the estimate of the nonlinear parameters. Nelson and Lewis [15] utilized Walling's approach together with the modified Gauss-Newton (Hartley) algorithm. Both report that for some examples the number of iterations required for convergence was less than that using a technique where all parameters were considered as nonlinear. Initial starting values for the nonlinear parameters were not considered by Nelson and Lewis and neither they nor Walling presented any comparison of computer time to convergence between their described methods and the methods using the conventional all nonlinear approach. In both the Walling and Nelson-Lewis methods the value of the linear parameters was calculated only after the estimate of the nonlinear parameters was made at any particular iteration; i.e., $\underline{\beta}^i$ remained constant for a given i^{th} iteration while searching for a correction factor to calculate $\underline{\alpha}^{i+1}$ utilizing a particular algorithm.

The same general method suggested by Nelson and Lewis was implied by Lawton and Sylvestre [10] in which they define a "reduced model" and apply the Hartley iterative technique to it to estimate the nonlinear parameters. Their reduced model is defined as $E(\underline{y}) = F\hat{\underline{\beta}}$ where $\hat{\underline{\beta}}$ is the vector of least square estimates of the linear parameters as defined in equation (6). It is not clear if their approach is mathematically the same as that of Nelson and Lewis, but it would appear to be. Four models are presented in which the same data are used to obtain a fit. In each case, their method reduces the number of iterations required to obtain convergence. Again, nothing is said of computer time necessary for convergence or starting vectors for the nonlinear parameters.

Papaioannou and Kempthorne [16] present an algorithm which minimizes a particular function with respect to linear and/or nonlinear parameters referred to as "parallel tangents and steepest descent." A suggestion is made to write the equation to be minimized in terms of nonlinear parameters only since $\hat{\underline{\beta}}$ can be written as $(F'F)^{-1}F'\underline{y}$. This reduces the dimensionality of the problem from m to q. They suggest that their method then be used to obtain $\underline{\alpha}^{min}$ which will allow in turn obtainment of $\underline{\beta}^{min}$. No examples of this suggested approach to the nonlinear regression problem are given. This is exactly what Spillman [21], Stevens [22], and Drane[2] each did for the exponential regression function. However, it would appear to be computer time consuming in comparison with other known methods. The cumbersomeness of this method was also demonstrated by Drane and Schucany [3]. Also, it would seem that stability problems might be more frequently encountered in this approach.

All of the above methods which recognize the relationships of the linear and nonlinear parameters have two primary advantages over techniques where all parameters are treated nonlinearly. These are:

1). The dimensionality of the vector of iterants and thus the number of initial guesses to be made is reduced from size $m = p + q$ (nonlinear) to size q (linear-nonlinear).

2). The sum-of-squared-error surface is simplified because of this dimensionality reduction. In some cases, faster convergence and less sensitivity to starting guesses is achieved. However, in many examples, which were run using the Nelson and Lewis [15] and Lawton and Sylvestre [10] methods, this was not found to be the case.

The algorithm presented in Chapter II for least squares estimation of nonlinear parameters, when some of the parameters are linear, makes use of the aforementioned fundamental approach, but with one additional criterion which greatly reduces the number of iterations required to reach convergence. In both the Hartley and Marquardt algorithms there are times within a given iteration when the error sum of squares must be calculated to ascertain the optimum correction terms to apply to the nonlinear parameters being estimated. In the methods described previously once the linear terms have been estimated for a given vector of nonlinear terms they are not calculated again within the iteration during the search for the "best" correction terms. This algorithm proposes calculating the linear parameter vector estimates within every iteration whenever the error sum of squares is required for obtainment of the best correction terms. This concept will become clear in Chapter II when the iterative scheme is presented in detail. Theory and examples using this concept are presented which will show, in many cases, a reduction in the number of iterations required to achieve convergence compared with other methods. Computer time required using this technique compared to other techniques also will be presented. All computational work was performed on the UNIVAC 1108 computer at the SMU Computing Center.

CHAPTER II

THEORETICAL BASIS FOR PARAMETER RECLASSIFICATION

The general theory of the nonlinear parameter reclassification approach as advocated in Chapter I will be presented using maximum-likelihood concepts (since, whenever $\underline{e} \sim N(\underline{0}, I\sigma^2)$ the least squares estimate of $\underline{\theta}$ is also the maximum likelihood estimate of $\underline{\theta}$). Richards [17] alludes to this approach, but only indirectly as it will be applied here. However, much of the following regarding maximum-likelihood estimation is from Richard's suggested approach to the problem. The theory will then be presented specifically for the iterative scheme as used in the given algorithm to obtain $\hat{\underline{\theta}}$ or $(\hat{\underline{\beta}}, \hat{\underline{\alpha}})$.

For this problem we will consider maximum-likelihood estimates which are asymptotically jointly normal and efficient. (The following results will be true under less stringent conditions, but not necessarily relevant.) Let $\tilde{\underline{\beta}}(\underline{\alpha})$ be the maximum-likelihood value of $\underline{\beta}$ for a given fixed $\underline{\alpha}$, and $\hat{\underline{\theta}}$, $\hat{\underline{\beta}}$, $\hat{\underline{\alpha}}$ be the maximum-likelihood values of $\underline{\theta}$, $\underline{\beta}$, and $\underline{\alpha}$, respectively. As defined previously, $\underline{\theta}$ is the $m \times 1$ vector of nonlinear parameters and $(\underline{\beta}, \underline{\alpha})$ are the linear-nonlinear parameter vectors as reclassified where $\underline{\beta}$ is $p \times 1$, $\underline{\alpha}$ is $q \times 1$, and $m = p + q$.

Let B be the matrix with elements b_{ij}, where

$$b_{ij} = \left. \frac{\partial \tilde{\beta}_i(\underline{\alpha})}{\partial \alpha_j} \right|_{\alpha = \hat{\alpha}} \qquad (i = 1, 2, \ldots, p;\ j = 1, 2, \ldots, q) \ .$$

Let $$L(\underline{\alpha}) = \ell\{\underline{\alpha},\underline{\tilde{\beta}}(\underline{\alpha})\} ,$$

where $$\ell(\underline{\Theta}) = \ell\{\underline{\alpha},\underline{\beta}\} = \log \prod_{h=1}^{n} p(\underline{x}_h;\underline{\Theta})$$

and $p(\underline{x};\underline{\Theta})$ is the pdf of $\underline{x}$ given $\underline{\Theta}$.

Let $$T_i(\underline{\alpha}) = \frac{\partial}{\partial\alpha_i} \ell\{\underline{\alpha},\underline{\tilde{\beta}}(\underline{\alpha})\} \qquad (i = 1, 2, \dots, q)$$

and $$\lambda_{ij} = -\left.\frac{\partial^2\ell(\Theta)}{\partial\Theta_i\partial\Theta_j}\right|\Theta = \hat{\Theta} \qquad (i,j = 1, 2, \dots, m) .$$

Let $$(\lambda_{ij}) = \Lambda = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \text{ and } \Lambda^{-1} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} ,$$

where M_{11}, L_{11} are $q \times q$ matrices, M_{12}, L_{12} are $q \times p$ matrices, M_{21}, L_{21} are $p \times q$ matrices, and M_{22}, L_{22} are $p \times p$ matrices.

Now $\underline{\tilde{\beta}}(\underline{\alpha})$ is a solution of the equations

$$\frac{\partial\ell(\underline{\Theta})}{\partial\beta_j} = 0 \qquad (j = 1, 2, \dots, p) . \tag{9}$$

Partial differentiation of equation (9) with respect to $\alpha_1,\alpha_2, \dots, \alpha_q$ gives

$$\frac{\partial^2\ell(\underline{\Theta})}{\partial\alpha_i\partial\beta_j} + \sum_{k=1}^{p} \frac{\partial^2\ell(\underline{\Theta})}{\partial\beta_k\partial\beta_j} \frac{\partial\tilde{\beta}_k(\underline{\alpha})}{\partial\alpha_i} = 0 \quad (i = 1,2,\dots, q;\ j = 1,2,\dots,p) .$$

Putting $\underline{\alpha} = \underline{\hat{\alpha}}$ in the above, we have in matrix notation

$$-M_{21} - M_{22}B = \phi$$

where ϕ is the $p \times q$ null matrix.

Solving for B we obtain

$$B = -M_{22}^{-1} M_{21} . \tag{10}$$

For $j = 1, 2, \dots, q$, we have

$$\frac{\partial L(\underline{\alpha})}{\partial\alpha_j} = \frac{\partial}{\partial\alpha_j} \ell\{\underline{\alpha},\underline{\tilde{\beta}}(\underline{\alpha})\} + \sum_{k=1}^{p} \frac{\partial}{\partial\beta_k} \ell\{\underline{\alpha},\underline{\tilde{\beta}}(\underline{\alpha})\} \frac{\partial\tilde{\beta}_k(\underline{\alpha})}{\partial\alpha_j} . \tag{11}$$

The second term on the right-hand side of equation (11) is zero by equation (9) and hence

$$\frac{\partial L(\underline{\alpha})}{\partial \alpha_j} = T_j(\underline{\alpha})$$

Also, for i,j = 1, 2, ..., q, we have

$$\frac{\partial^2 L(\underline{\alpha})}{\partial \alpha_i \partial \alpha_j} = \frac{\partial^2 \ell(\underline{\theta})}{\partial \theta_i \partial \theta_j} + \sum_{k=1}^{p} \frac{\partial^2 \ell(\underline{\theta})}{\partial \theta_k \partial \theta_j} \frac{\partial \tilde{\beta}_k(\underline{\alpha})}{\partial \alpha_i}$$

and putting $\underline{\alpha} = \hat{\underline{\alpha}}$ in the above, we have in matrix notation

$$\frac{\partial^2 L(\hat{\underline{\alpha}})}{\partial \alpha_i \partial \alpha_j} = -M_{11} - M_{12} B$$

by substitution of equation (10) for B,

$$= -M_{11} + M_{12} M_{22}^{-1} M_{21}$$

and by properties of partioned matrices and their inverses,

$$= -L_{11}^{-1} \quad . \tag{12}$$

By the same properties,

$$B = -M_{22}^{-1} M_{21} = L_{21} L_{11}^{-1} \tag{13}$$

and

$$L_{22} = M_{22}^{-1} + L_{21} L_{11}^{-1} L_{12} \tag{14}$$

The above results show that the complete maximum-likelihood estimate, $(\hat{\underline{\theta}})$, together with its asymptotic covariance matrix estimated by Λ^{-1}, may be obtained by the following method:

(i) Obtain $\tilde{\beta}_1, \tilde{\beta}_2, \ldots, \tilde{\beta}_p$ as functions of the unknowns $\alpha_1, \alpha_2, \ldots, \alpha_q$.

(ii) Substitute for $\beta_1, \beta_2, \ldots, \beta_p$ in the likelihood function ℓ[or in $\partial\ell/\partial\alpha_i (i=1,2,\ldots,q)$] the functions from (i) to obtain a modified likelihood function $L(\alpha_1, \alpha_2, \ldots, \alpha_q)$ and hence the modified maximum-likelihood equations $T_i(\alpha_1, \alpha_2, \ldots, \alpha_q) = 0$ $(i = 1, 2, \ldots, q)$.

(iii) Using these modified equations proceed as if performing an ordinary maximum-likelihood estimation procedure for $\hat{\underline{\alpha}}$, finding also the corresponding estimated information matrix $(-\partial^2 L(\hat{\alpha})/\partial\alpha_i\partial\alpha_j)$ and its inverse. Equation (12) shows that this will give L_{11}, which is in fact the estimated covariance matrix of $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_q$.

(iv) From $\hat{\underline{\alpha}}$ we can now compute B, $\hat{\underline{\beta}} = \tilde{\underline{\beta}}(\hat{\underline{\alpha}})$, M_{22}^{-1}, and hence, using equations (13) and (14), we may obtain L_{12} and L_{22} to complete the estimated covariance matrix Λ^{-1}.

The equations $T_i(\underline{\alpha}) = 0$ are difficult to solve in most instances; therefore, an iterative procedure must be employed. The idea is to guess an initial value of $\underline{\alpha}$, say $\underline{\alpha}^0$, and then essentially proceed through steps i-iv to obtain $\underline{\alpha}^1$ and thus $\underline{\beta}^1$ $[=\tilde{\beta}(\underline{\alpha}^1)]$ using a particular algorithm (Gauss-Newton, steepest descent, etc.) Using $\underline{\alpha}^1$ this procedure is continued until termination at some $(\underline{\alpha}^{**}, \underline{\beta}^{**})$ which satisfies some pre-determined criterion such as the sum of squared error converging to the same value. The exact method advocated in this dissertation will become clear in the following presentation of the theoretical iterative approach.

The modified Gauss-Newton method (Hartley) and the combination of this method with the method of steepest descent (Marquardt) both seek to minimize the error sum of squares with respect to the unknown nonlinear parameters by the method of least squares. The proposed algorithm makes use of the concept of reclassifying the nonlinear parameters into linear and nonlinear ones and employing the above two techniques for estimation of the nonlinear parameters. The method needs as input:

1). A set of observations of the independent variables $\underline{x}_h$ and dependent variable y_h.

2). A subroutine which will read these data into the computer in a specified manner.

3). An algorithm that computes the partial derivatives of the desired regression function(the program has an option which will bypass this algorithm and use approximate partial derivatives obtained by difference quotients if desired).

4). Initial guesses of the parameter vector which will minimize the least squares function.

The algorithm allows the option of considering all the parameters as nonlinear or reclassifying them as linear-nonlinear. Thus, the dimension of the parameter vector in item 4 above will vary according to the option exercised in the program.

The procedure minimizes the function

$$Q(\underline{\Theta}) = \sum_{h=1}^{n} [y_h - f(x_{h1}, x_{h2}, \ldots, x_{hk}; \Theta_1, \Theta_2, \ldots, \Theta_m)]^2 \qquad (15)$$

where

n = number of observations

y_i = dependent variable of the i^{th} observation

x_{ij} = j^{th} independent variable of the i^{th} observation

f = function to be fit

Θ_i = i^{th} parameter of f

$Q(\underline{\Theta})$ = a function of the m - tuple, $\underline{\Theta} = (\Theta_1, \Theta_2, \ldots, \Theta_m)'$

When $\underline{\Theta}$ is considered as all nonlinear the program takes the initial guess $\underline{\Theta}^0$ and computes the necessary corrections to $\underline{\Theta}^0$ to obtain $\underline{\Theta}^{min}$, the m - tuple which minimizes $Q(\underline{\Theta})$. The corrections are computed by substituting a first order Taylor approximation of $f(\underline{\Theta})$ into equation (15) and forming the least squares equations $Q_i(\underline{\Theta}) = 0$ for i = 1, 2, ..., m, where $Q(\underline{\Theta})$ is written as

$$Q(\underline{\Theta}) = \sum_{h=1}^{n} [y_h - f(\underline{x}_h;\underline{\Theta})]^2 \qquad (16)$$

and $Q_i(\underline{\theta})$ denotes the partial derivative of $Q(\underline{\theta})$ with respect to θ_i. Substituting the first order Taylor approximation, equation (16) becomes

$$Q(\underline{\theta}) = \sum_{h=1}^{n} [y_h - f(\underline{x}_h;\underline{\theta}^0) - \sum_{j=1}^{m} f_j(\underline{x}_h;\underline{\theta}^0)\delta_j]^2 \qquad (17)$$

where if f is a function of $(\theta_1, \theta_2, \ldots, \theta_m)$, then $f_i = \frac{\partial f}{\partial \theta_i}$ and $f_{ij} = \frac{\partial^2 f}{\partial \theta_i \partial \theta_j}$, the first and second partial derivatives of f with respect to θ_i.

Also, $\theta_i^{\ k}$ = i^{th} component of k^{th} θ vector in the sequence computed by the program starting with $\underline{\theta}^0$

$\delta_i^{\ k} = \theta_i^{\ k+1} - \theta_i^{\ k} = i^{th}$ component of $\underline{\delta}^k$,

and $\underline{x}_i$ = k - tuple $(x_{i1}, x_{i2}, \ldots, x_{ik})$.

Differentiating equation (17) with respect to θ_i we obtain

$$Q_i(\underline{\theta}) = -2\sum_{h=1}^{n} [y_h - f(\underline{x}_h;\underline{\theta}^0) - \sum_{j=1}^{m} f_j(\underline{x}_h;\underline{\theta}^0)\delta_j] [f_i(\underline{x}_h;\underline{\theta}^0)] \ . \qquad (18)$$

(In this regard, equation A-4 is incorrect in reference (5) and should be replaced with equation (18) above.) Setting $Q_i(\underline{\theta}) = 0$ in equation (18), we obtain

$$\sum_{h=1}^{n} \sum_{j=1}^{m} f_i(\underline{x}_h;\underline{\theta}^0) f_j(\underline{x}_h;\underline{\theta}^0)\delta_j = \sum_{h=1}^{n} [y_h - f(\underline{x}_h;\underline{\theta}^0)] f_i(\underline{x}_h;\underline{\theta}^0) \qquad (19)$$

which can be solved for m values of δ_j . Once the values of δ_j have been computed, the algorithm utilizes a particular method (depending on whether the Hartley or Marquardt technique is employed) to determine the optimum magnitude of the correction.

Equation (19) can be written in matrix notation as

$$A\ \underline{\delta} = \underline{\nu} \quad , \qquad (20)$$

where $\underline{\delta} = (\delta_1, \delta_2, \ldots, \delta_m)'$, $\underline{\nu} = \Bigg(\sum_{h=1}^{n} [y_h - f(\underline{x}_h;\underline{\theta}^0)]\ f_1(\underline{x}_h;\underline{\theta}^0)$,

$$\sum_{h=1}^{n} [y_h - f(\underline{x}_h;\underline{\theta}^0)]\ f_2(\underline{x}_h;\underline{\theta}^0),\ \ldots,\ \sum_{h=1}^{n} [y_h - f(\underline{x}_h;\underline{\theta}^0)]\ f_m(\underline{x}_h;\underline{\theta}^0)\Big),$$ and

A is an m × m matrix whose terms a_{jk} are given by

$$a_{jk} = \sum_{i=1}^{n} f_j(\underline{x}_h;\underline{\theta}^0)\ f_k(\underline{x}_h;\underline{\theta}^0)\quad .$$

Equation (15) can be written in the following form when reclassifying the nonlinear parameters as linear-nonlinear:

$$Q(\underline{\beta},\underline{\alpha}) = \sum_{h=1}^{n} [y_h - \sum_{j=1}^{p} \beta_j g_j(x_{h1},x_{h2},\ldots,x_{hk};\ \alpha_1,\alpha_2,\ldots,\ \alpha_q)]^2 \quad (21)$$

where n, y_i, and x_{ij} are as before and

$\sum_{j=1}^{p} \beta_j g_j$ = function to be fit

β_i = i^{th} linear parameter of the function to be fit

α_i = i^{th} nonlinear parameter of the function to be fit

$Q(\underline{\beta},\underline{\alpha})$ = a function of the p-tuple $(\beta_1,\ \beta_2,\ \ldots,\ \beta_p)$ and q-tuple $(\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_q)$.

The assumption is made that $1 \leq p < m$ of the m parameters $(\underline{\theta})$ are linear and $m - p = q$ are nonlinear.

When $\underline{\theta}$ is reclassified as linear-nonlinear $(\underline{\beta},\underline{\alpha})$ the program requires only that an initial estimate of the nonlinear parameters, say $\underline{\alpha}^0$, be provided. The function to be minimized then becomes

$$Q(\underline{\beta}|\underline{\alpha}^0) = \sum_{h=1}^{n} [y_h - \sum_{j=1}^{p} \beta_j g_j\ (x_{h1},x_{h2},\ldots,x_{hk};\ \alpha_1^{\ 0},\alpha_2^{\ 0},\ldots,\alpha_q^{\ 0}))]^2 \quad (22)$$

The least squares estimate of $\underline{\beta}$ is obtained in the usual manner of differentiating equation (22) with respect to β_j and setting the result equal to zero. This gives p linear equations in p unknowns which can readily be solved for $\underline{\beta}^0$. In matrix notation

$$\underline{\beta}^0 = (F'F)^{-1} F'\underline{y} \tag{23}$$

where the notation is the same as given for equation (8). Based upon this $\underline{\beta}^0$ we wish to form a better choice for the nonlinear parameters $\underline{\alpha}$ than our initial guessed estimate $\underline{\alpha}^0$. The program utilizes the same nonlinear approach described previously for $\underline{\Theta}$ to obtain the correction to $\underline{\alpha}^0$ which will provide a better estimate of $\underline{\alpha}$, say $\underline{\alpha}^1$. Equation (20) can be partioned in the following manner having reclassified the nonlinear parameters as linear-nonlinear

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \underline{\delta}_1 \\ \underline{\delta}_2 \end{bmatrix} = \begin{bmatrix} \underline{\nu}_1 \\ \underline{\nu}_2 \end{bmatrix} \tag{24}$$

where

$\underline{\delta}_1 = (\delta_1, \delta_2, \ldots, \delta_p)'$, the difference between the (k+1) and k^{th} iteration value for the $\underline{\beta}$ linear parameters

$\underline{\delta}_2 = (\delta_1, \delta_2, \ldots, \delta_q)'$, the difference between the (k+1) and k^{th} iteration value for the $\underline{\alpha}$ nonlinear parameters

$$\underline{\nu}_1 = \Bigg(\sum_{h=1}^{n} [y_h - \sum_{j=1}^{p} \beta_j{}^0 g_j(\underline{x}_h;\underline{\alpha}^0)] F_1(\underline{x}_h;\underline{\beta}^0,\underline{\alpha}^0),\ \sum_{h=1}^{n} [y_h - \sum_{j=1}^{p} \beta_j{}^0 g_j(\underline{x}_h;\underline{\alpha}^0)] F_2(\underline{x}_h;\underline{\beta}^0,\underline{\alpha}^0),$$
$$\ldots, \sum_{h=1}^{n} [y_h - \sum_{j=1}^{p} \beta_j{}^0 g_j(\underline{x}_h;\underline{\alpha}^0)] F_p(\underline{x}_h;\underline{\beta}^0,\underline{\alpha}^0)\Bigg)'$$

$$\underline{\nu}_2 = \Bigg(\sum_{h=1}^{n} [y_h - \sum_{j=1}^{p} \beta_j{}^0 g_j(\underline{x}_h;\underline{\alpha}^0)] F_1^*(\underline{x}_h;\underline{\beta}^0,\underline{\alpha}^0),\ \sum_{h=1}^{n} [y_h - \sum_{j=1}^{p} \beta_j{}^0 g_j(\underline{x}_h;\underline{\alpha}^0)] F_2^*(\underline{x}_h;\underline{\beta}^0,\underline{\alpha}^0),$$
$$\ldots, \sum_{h=1}^{n} [y_h - \sum_{j=1}^{p} \beta_j{}^0 g_j(\underline{x}_h;\underline{\alpha}^0)] F_q{}^*(\underline{x}_h;\underline{\beta}^0,\underline{\alpha}^0)\Bigg)'$$

and A is the previously described m × m matrix which has been partioned as shown where A_{11} is p × p, A_{12} is p × q, A_{21} is q × p, and A_{22} is q × q and where if F is a function of $(\underline{\beta},\underline{\alpha})$ equal to $\sum_{j=1}^{n} \beta_j{}^0 g_j(\underline{x}_h;\underline{\alpha}^0)$, then $F_i = \frac{\partial F}{\partial \beta_i}$ and $F_j{}^* = \frac{\partial F}{\partial \alpha_j}$, the first derivatives of F with respect to β_i and α_j for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, q$.

Equation (24) can be written as

$$A_{11}\underline{\delta}_1 + A_{12}\underline{\delta}_2 = \underline{\nu}_1 \tag{25}$$

$$A_{21}\underline{\delta}_1 + A_{22}\underline{\delta}_2 = \underline{\nu}_2 \tag{26}$$

But $\underline{\nu}_1 = \underline{0}$ by our choice of the least squares estimate of $\underline{\beta}^0$ given $\underline{\alpha}^0$. Thus, equations (25) and (26) become

$$A_{11}\underline{\delta}_1 + A_{12}\underline{\delta}_2 = \underline{0}$$

$$A_{21}\underline{\delta}_1 + A_{22}\underline{\delta}_2 = \underline{\nu}_2$$

which can be solved for $\underline{\delta}_2$ to give

$$\begin{aligned}\underline{\delta}_2 &= (A_{22} - A_{21}A_{11}{}^{-1}A_{12})^{-1}\ \underline{\nu}_2 \\ &= A^{*-1}\ \underline{\nu}_2\end{aligned} \tag{27}$$

Equation (27) is used by the program at each iteration to obtain the correction terms for the nonlinear parameters. This equation essentially replaces equation (20) which is used when all the parameters are nonlinear. The program allows the option of using the Hartley or Marquardt technique to obtain the solution to equation (27).

Hartley's algorithm corrects $\underline{\alpha}$ by only a multiple of $\underline{\delta}_2$, say $r\underline{\delta}_2$, where $-\infty \leq r \leq \infty$ within an iteration to give the minimum value for the error sum of squares, $Q(\underline{\alpha}|\underline{\beta})$. The value of r is not restricted to the

[0,1] interval in this particular program as it is in Hartley's original algorithm. The program calculates $\underline{\beta}$ for a number of values of r in arriving at the value of r which minimizes $Q(\underline{\alpha}|\underline{\beta})$.

The algorithm of Marquardt involves the system of linear equations

$$(A^* + \lambda I)\underline{\delta}_2 = \underline{\nu}_2 \tag{28}$$

where λ is a scalar ≥ 0, I is the identity matrix, and the solution $\underline{\delta}_2$ gives the required steps to the next iterant. When $\lambda = 0$, equation (28) is identical to the normal equations (27) of the Gauss-Newton method and yields the value of $\underline{\delta}_2$ which determines the exact minimum of $Q(\underline{\alpha}|\underline{\beta})$; whereas, on the other hand, as $\lambda \to \infty$, $\underline{\delta}_2$ becomes more nearly proportional to $\underline{\nu}_2$ which (apart from a positive constant) is the negative gradient or vector of steepest descent of $Q(\underline{\alpha}|\underline{\beta})$ at $\underline{\beta}^0$.

In summary, the iteration scheme is

1). Initialize $\underline{\alpha}$, call it $\underline{\alpha}^0$.

2). Calculate $\underline{\beta}^0 = (F'F)^{-1}F'\underline{y} = A_{11}^{-1}F'\underline{y}$.

3). Calculate $\underline{\delta}_2^{\,0} = A^{*-1}\ \underline{\nu}_2$ using the Hartley or Marquardt technique to obtain $\underline{\alpha}^1$ (calculating $\underline{\beta}^0$ each time a trial value of $\underline{\alpha}^1$ is required).

4). Check for convergence; if yes, proceed to step 2, and stop. If no, proceed to step 2, and continue.

CHAPTER III

CONVERGENCE CRITERIA AND CONFIDENCE REGIONS

Under certain conditions the sequence $\{\underline{\Theta}^i\}$ converges to $\underline{\Theta}^{min}$. Conditions for convergence when $\underline{\Theta}$ is considered as all nonlinear using the Taylor series method are discussed in detail by Hartley [7]. In summary, these conditions for $f(\underline{x}_h;\underline{\Theta})$ are:

1). f_i and f_{ij} exist and are continuous in $\underline{\Theta}$ for all $\underline{x}_h$; $i,j = 1, 2, \ldots, m$, where f_i and f_{ij} are as defined in equation (17).

2). There exists a bounded convex set S such that for every $\underline{\Theta}\varepsilon S$ and every nontrivial finite set $\{u_i\}$ with $\sum_{i=1}^{m} u_i^2 > 0$, $\sum_{h=1}^{n}\left[\sum_{i=1}^{m} u_i f_i(\underline{x}_h;\underline{\Theta})\right]^2 > 0$. This assumption will assure that the matrix of first partial derivatives has an inverse.

3). There exists a $\underline{\Theta}^0$ in the interior of S such that $Q(\underline{\Theta}^0) < \bar{Q}$ where $\bar{Q} = \lim_{\bar{S}} \inf Q(\underline{\Theta})$ and $\bar{S}$ = complement of S. This assumption guarantees that the search will stop within the area where assumption 2 holds and that it will converge to a point in S.

When $\underline{\Theta}$ is reclassified as linear-nonlinear $(\underline{\beta},\underline{\alpha})$, conditions for convergence for the function $\sum_{j=1}^{p} \beta_j g_j(\underline{x}_h;\underline{\alpha})$ using the Hartley technique are:

1). F_i and F_j^* exist and are continuous in $(\underline{\beta},\underline{\alpha})$ for all $\underline{x}_h$; $i = 1, 2, \ldots, p$; $j = 1, 2, \ldots, q$, where F_i and F_j^* are as defined following equation (24).

2). There exists a bounded convex set W such that for every $(\underline{\beta},\underline{\alpha})\varepsilon W$ and every nontrivial finite set $\{u_i\}$ and $\{u_j^*\}$ with $\left(\sum_{i=1}^{p} u_i^2 + \sum_{j=1}^{q} u^*{}_j^2\right) > 0$,

$$\sum_{h=1}^{n} \left[\sum_{i=1}^{p} u_i F_i(\underline{x}_h;\underline{\beta},\underline{\alpha}) + \sum_{j=1}^{q} u_j^* F_j^*(\underline{x}_h;\underline{\beta},\underline{\alpha})\right]^2 > 0 \ . \tag{29}$$

3). There exists a $\underline{\alpha}^0$ in the interior of W such that $Q(\underline{\alpha}^0|\underline{\beta}^0) < \bar{\bar{Q}}$ where $\bar{\bar{Q}} = \lim_{\bar{W}} \inf Q(\underline{\alpha}|\underline{\beta})$ and $\bar{W}$ = complement of W.

Starting with the vector $\underline{\alpha}^0$ in assumption 3) above, $\underline{\beta}^0 = (F'F)^{-1}F'\underline{y}$ is computed. Because of assumption 2) the determinant of linear equations $(F'F)\underline{\beta}^0 = F'\underline{y}$ has rank p and thus can always be solved for the p × 1 vector $\underline{\beta}^0$. The corrections to the starting vector $\underline{\alpha}^0$ are then computed. These corrections are given by equation (27).

By a well known result in matrix algebra, $A^* = A_{22} - A_{21}A_{11}^{-1}A_{12}$ is non-singular since A_{11} and A are non-singular by assumption 2. Therefore, we can say the determinant of linear equations $A^*\underline{\delta}_2 = \underline{\nu}_2$ has rank q and can thus be solved for the q × 1 vector $\underline{\delta}_2$.

Now consider the function

$$\check{Q}(r) = Q(\underline{\alpha}^0 + r\underline{\delta}_2) \text{ for } 0 \leq r \leq 1 \tag{30}$$

and denote by r' the value of r for which Q(r) is a minimum on the interval $0 \leq r \leq 1$. (The implication of r outside this interval for this algorithm will be discussed later.) Let $\underline{\alpha}^1 = \underline{\alpha}^0 + r'\underline{\delta}_2$. Then

$$Q(\underline{\alpha}^1) \leq Q(\underline{\alpha}^0) < \bar{\bar{Q}} \tag{31}$$

so that $\underline{\alpha}^1$ lies in the interior of W.

The above computation is repeated at $\underline{\beta}^1$ and $\underline{\alpha}^1$, etc. There results a sequence of vectors $(\underline{\beta}^t,\underline{\alpha}^t)$, t = 1, 2, ..., all within the convex set W with $\underline{\beta}^t$ being the least squares solution given some $\underline{\alpha}^t$ and we can say

$$\lim_{t \to \infty} Q(\underline{\alpha}^t) = Q^{**} \quad . \tag{32}$$

Consider a point of accumulation $\underline{\alpha}^{**}$ of this bounded sequence and subsequence $\underline{\alpha}^s$ with

$$\lim_{s \to \infty} \underline{\alpha}^s = \underline{\alpha}^{**} \tag{33}$$

Since

$$\lim_{s \to \infty} Q(\underline{\alpha}^s) = Q(\underline{\alpha}^{**}) \leq Q(\underline{\alpha}^0) < \bar{\bar{Q}} \tag{34}$$

it follows from assumption 3 that $\underline{\alpha}^{**}$ must be an interior point of W. We shall now show that at this limit point, $\underline{\alpha}^{**}$, the first partials, $Q_i(\underline{\alpha}^{**})$ for i = 1, 2, ..., q, must all be zero.

Let the ${}_i\delta^{**}$ be the solution to equation (27); i.e.,

$$A^*\underline{\delta}^{**} = \underline{\nu}_2 \quad , \tag{35}$$

where A* and $\underline{\nu}_2$ are evaluated at $\underline{\alpha}^{**}$.

Because of the continuity assumptions and equation (29)

$$\lim_{s \to \infty} {}_i\delta^s = {}_i\delta^{**} \quad . \tag{36}$$

Further, from equations (35) and (29)

$$\sum_{i=1}^{q} Q_i(\underline{\alpha}^{**})\ {}_i\delta^{**} = -2 \sum_{h=1}^{n} \{f_i(\underline{x}_h;\underline{\alpha}^{**})\ {}_i\delta^{**}\}^2 < 0, \tag{37}$$

provided

$$\sum_{i=1}^{q} {}_i\delta^{**2} > 0 \quad .$$

But equation (37) implies that the differential of Q in the direction proportional to the ${}_i\delta^{**}$ is negative. Therefore, assuming that $\sum_{i=1}^{q} {}_i\delta^{**2}$ is strictly positive, it follows from equation (36) that for all $\underline{\alpha}^s$ in a small neighborhood of $\underline{\alpha}^{**}$ the differential of Q in the direction proportional to the ${}_i\delta^s$ would be smaller than a fixed quantity, say e. Since the second differential of Q in these directions (this differential may be defined as the differential with regard to the variable r as defined by equation (30) with $\underline{\alpha}^s$ replacing $\underline{\alpha}^0$) is bounded over a unit distance by a bound, call it M, it follows that the minimum of Q in the direction of ${}_i\delta^s$ must be below $Q(\underline{\alpha}^t)$ by at least the amount $er - \frac{1}{2}Mr^2$ where r is the fractional distance moved in the direction proportional to ${}_i\delta^s$ from $\underline{\alpha}^s$. Choosing $r^* = \min(1, \frac{|e|}{M})$ the minimum of Q in the direction proportional to ${}_i\delta^s$ would all be below $Q(\underline{\alpha}^s)$ by at least the amount $\frac{1}{2}$ er*. This contradicts equation (32) which states that the $Q(\underline{\alpha}^t)$ of the original sequence t converges to Q** which also would be the limit of the subsequence $Q(\underline{\alpha}^s)$. Thus, a contradiction is reached unless $\sum_{i=1}^{q} {}_i\delta^{**2} = 0$ which implies, because of the full rank of equation (35), that

$$\sum_{i=1}^{q} Q_i^{\,2}(\underline{\alpha}^{**}) = 0 \ .$$

Therefore, a subsequence $\underline{\alpha}^s$ of the sequence $\underline{\alpha}^t$ converges to a solution $\underline{\alpha}^{**}$ of the least squares equations

$$Q_i(\underline{\alpha}^{**}) = 0 \qquad (i = 1, 2, \ldots, q) \ .$$

The original sequence $\underline{\alpha}^t$ will converge to $\underline{\alpha}^{**}$ for almost all problems since if there were an infinite subsequence of $\underline{\alpha}^{s'}$ not converging to $\underline{\alpha}^{**}$, then a subsequence of these, $\underline{\alpha}^{s''}$ would tend to a limit $\underline{\alpha}^{***} \neq \underline{\alpha}^{**}$. Then $Q(\underline{\alpha}^{**}) = Q(\underline{\alpha}^{***})$ and $\underline{\alpha}^{***}$ must be a stationary point. It is highly improbable that there would be a regression surface and set of observations, $\underline{x}_h$ and y_h , such that Q has two stationary points yielding precisely the same value of Q. The above convergence proof rests on this assumption that Q has no two stationary points yielding the same values of Q.

The above convergence proof for the Hartley technique using the linear-nonlinear approach is presented with r on the [0,1] interval since this is the interval originally chosen by Hartley and in widespread use. As noted previously, this algorithm allows r to range over the $(-\infty, +\infty)$ interval, but in a somewhat restricted sense. There is a particular criterion in the program which governs its final value for any given iteration as it is allowed to vary. The range of r for this program together with the method employed for its calculation does not alter the assumptions made on $f(\underline{x}_h;\underline{\theta})$ or $\sum_{j=1}^{p} \beta_j g_j(\underline{x}_h;\underline{\alpha})$ to obtain convergence, nor does it change the above proof to any great extent.

The algorithm of Marquardt when $\underline{\theta}$ is considered as all nonlinear involves the system of linear equations given by

$$(A + \lambda I)\underline{\delta} = \underline{\nu}$$

where A, $\underline{\delta}$, and $\underline{\nu}$ are as defined in equation (20), λ is a scalar ≥ 0 and I is the identity matrix. The theoretical basis for this algorithm was presented by Marquardt [11] in three theorems. Meeter [14] gives an alternate and more general proof of Marquardt's Theorem 1 and discusses its implication on Marquardt's Theorem 2.

Marquardt's algorithm as applied to $\underline{\theta}$ when reclassified as linear-nonlinear $(\underline{\beta},\underline{\alpha})$ involves the system of linear equations given by

$$(A^{**} + \lambda I)\underline{\delta}_2 = \underline{\nu}_2 \qquad (38)$$

where A*, $\underline{\delta}_2$, and $\underline{\nu}_2$ are as defined in equation (27) and λ,I are as defined previously. The three theorems and their proofs remain the same as for $\underline{\theta}$ considered as all nonlinear when A*, $\underline{\delta}_2$, and $\underline{\nu}_2$ are substituted, respectively, for A, $\underline{\delta}$, and $\underline{\nu}$. Therefore, the theorems and their proofs will not be repeated here. Marquardt discusses convergence proofs in his paper [11]. His discussion is applicable here with the above substitutions made for A, $\underline{\delta}$, and $\underline{\nu}$.

As pointed out by Draper and Smith [4], an idea of the nonlinearity in the model can be obtained after estimating $\underline{\theta}$ (or $\underline{\alpha}$ in the case of parameter reclassification) by evaluating the ellipsoidal confidence region obtained on the assumption that the linearized form of the model is valid around $\hat{\underline{\theta}}(\hat{\underline{\alpha}})$, the final least squares estimate of $\underline{\theta}(\underline{\alpha})$. These are given by the following expressions:

$$(\underline{\theta}-\hat{\underline{\theta}})'\hat{A}'\hat{A}(\underline{\theta}-\hat{\underline{\theta}}) \le \frac{mQ(\hat{\underline{\theta}})}{n-m} F(m, n-m, 1-\alpha) \qquad (39)$$

for the all nonlinear approach, and

$$(\underline{\alpha}-\hat{\underline{\alpha}})'\hat{A}^{*\prime}\hat{A}^{*}(\underline{\alpha}-\hat{\underline{\alpha}}) \le \frac{qQ(\hat{\underline{\alpha}}|\hat{\underline{\beta}})}{n-m} F(q, n-m, 1-\alpha) \qquad (40)$$

for the linear-nonlinear approach, where $\hat{A}$ and $\hat{A}^*$ are evaluated at $\hat{\underline{\theta}}$ and $\hat{\underline{\alpha}}$ (implying at $\hat{\underline{\beta}}$ also), respectively, F(m, n-m, 1-α) is the 1-α point (upper α-point) of the F distribution with m, n-m degrees of freedom.

When the difference between successive values $\underline{\theta}^{i+1}$ and $\underline{\theta}^{i}$ ($\underline{\alpha}^{i+1}$ and $\underline{\alpha}^{i}$ in the reclassification approach) is sufficiently small so that the linearization procedure terminates with $\underline{\theta}^{i+1} = \hat{\underline{\theta}}$ ($\underline{\alpha}^{i+1} = \hat{\underline{\alpha}}$), then $Q(\hat{\underline{\theta}})(Q(\hat{\underline{\alpha}}))$ is a minimum value of $Q(\underline{\theta})$ $(Q(\underline{\alpha}))$ to the accuracy imposed by the termination procedure selected. The ellipsoid above will not be a true confidence region when the model is nonlinear.

An exact confidence contour is defined by taking $Q(\underline{\theta})$ in the all nonlinear case or $Q(\underline{\alpha})$ in the linear-nonlinear case to be a constant, but since the correct distribution properties in the general nonlinear case are not known one is not able to obtain a specified probability level. However, the contour can be chosen such that

$$Q(\underline{\theta}) = Q(\hat{\underline{\theta}})\{1 + \frac{m}{n-m} F(m, n-m, 1-\alpha)\} \tag{41}$$

for the all nonlinear approach, and

$$Q(\underline{\alpha}) = Q(\hat{\underline{\alpha}})\{1 + \frac{q}{n-m} F(q, n-m, 1-\alpha)\} \tag{42}$$

for the linear-nonlinear approach which will give an approximate 100(1-α)% confidence contour in the nonlinear case (equation (41) would provide an exact 100(1-α)% ellipsoidal confidence contour if the model were linear).

While suitable comparisons of mean squares can still be made visually, the usual F-tests for regression and lack of fit are not valid, in general, in the nonlinear case.

Measures of nonlinearity suggested by Beale[1] and discussed by Guttman and Meeter [6] can be used to help decide when linearized results provide acceptable approximations. Beale defines a theoretical measure of nonlinearity and intrinsic nonlinearity, N_Θ and N_ϕ, together with corresponding empirical measures of these quantities, $\hat{N}_\Theta$ and $\hat{N}_\phi$. He shows (extensions of Beale's work being made here by the author to the linear-nonlinear approach) that if N_ϕ is not too large equations (39) and (40) are confidence regions for $\underline{\Theta}$ and $\underline{\alpha}$, respectively, with associated probability greater than or equal to 1-α if the right-hand side of equation (39) is multiplied by

$$1 + \frac{n}{n-1} N_\phi \qquad (m = 1)$$

$$1 + \frac{n(m+2)}{(n-m)m} N_\phi \qquad (m \geq 2)$$

and if the right-hand side of equation (40) is multiplied by

$$1 + \frac{n}{n-1} N_\phi \qquad (q = 1)$$

$$1 + \frac{n(q+2)}{(n+q)q} N_\phi \quad (q \geq 2) \quad .$$

Hartley [8] presents a method that allows the construction of exact confidence regions for $\underline{\Theta}$ in the general case of nonlinear regression. The analysis rests on the fact that one can decompose the error sum of squares into two quadractic forms known as regression sum of squares of rank m and residual sum of squares of rank n-m and that these two forms will be independently distributed as $\sigma^2\chi^2$ for m and n-m degrees of freedom (by Cochran's theorem), thus providing an exact 100 $(1-\alpha)$% confidence region for $\underline{\Theta}$ as

$$\frac{\text{regression sum of squares}}{\text{residual sum of squares}} \leq \frac{m}{n-m} F(m,n-m,1-\alpha) \quad .$$

However, to be valid the above procedure requires the decomposition of the error sum of squares such that $f(\underline{x}_h;\underline{\Theta})$ is represented approximately as a m-term linear form of parameter functions $\omega_i(\Theta)$; i.e.,

$$f(\underline{x}_h;\underline{\Theta}) = \sum_{i=1}^{m} \omega_i(\Theta)\, Z_{hi}$$

where $\omega_i(\Theta)$ are a reparameterization of the Θ_i and Z_{hi} form a n × m matrix of rank m which does not depend on $\underline{\Theta}$. This method is not unique as it not only depends on the function to be fit but also on the method of linearization used. There is also the possibility that the resulting confidence region may be useless depending on the choice for the linear approximation of $f(\underline{x}_h;\underline{\Theta})$. Seibert [19] suggested a measure of efficiency of Hartley's exact confidence regions. Williams [24] also discusses the problem of exact confidence regions in nonlinear regression. His approach is essentially that of Hartley's (linearization of $f(\underline{x}_h;\underline{\theta})$), but not on as general a scale. Both methods provide exact confidence regions, but only after some simplifying assumptions on the function to be fit.

CHAPTER IV

EXAMPLES, DISCUSSION OF RESULTS, AND CONCLUSIONS

Many example problems were performed using the algorithm as described. The results of four of these models for one particular data set each, as well as two of the same models for 25 data sets each, will be presented to afford insight into the effectiveness of the new method when reclassifying the parameters to be estimated from all nonlinear to linear-nonlinear. These problems show quite markedly the variation in results which can be obtained depending on the model to be fit, the observed data, and the starting parameter vector considered.

The four models to be discussed are:

Model No.	Function	
	All Nonlinear	Linear-Nonlinear
1	$\Theta_1 e^{\Theta_2 x}$	$\beta_1 e^{\alpha_1 x}$
2	$\Theta_1+\Theta_2 e^{\Theta_3 x}$	$\beta_1+\beta_2 e^{\alpha_1 x}$
3	$\Theta_1 e^{\Theta_2 x}+\Theta_3 e^{\Theta_4 x}$	$\beta_1 e^{\alpha_1 x}+\beta_2 e^{\alpha_2 x}$
4	$\Theta_3 f_1+\ldots+\Theta_{12} f_{10}$	$\beta_1 f_1+\ldots+\beta_{10} f_{10}$

where, in Model 4, $f_1 = x_2 f$, $f_2 = x_3 f$, ..., $f_{10} = x_{11} f$ and $f = 1/(1 + \exp[\Theta_1(x_1+\Theta_2)])$ for all nonlinear and $f = 1/(1 + \exp[\alpha_1(x_1+\alpha_2)])$ for linear-nonlinear. In addition, x_1 is a continuous variable and x_2, ..., x_{11} are all 0 - 1 variables.

The following data were used in the comparison of the four models using one set of data each: The data used for Model 1 was the same as that used by Lawton and Sylvestre [10]. Data for Model 2 was taken from a fertilizer experiment described by Hartley [7]. Data for Models 3 and 4 were from experiments performed on blood plasma and cats, respectively, and were furnished by Drs. G. T. Shires and John Dietschy, University of Texas Southwestern Medical School, and Dr. Barbara Kent, Emory University School of Medicine.

Table 1 presents the results of fitting the above four models with the data described above using both the Hartley and Marquardt techniques when considering the parameters as all nonlinear and linear-nonlinear. Table 2 presents the same type information for Models 2 and 3 when more than one data set was considered. Data used to obtain the Table 2 results for Model 2 were from experiments reported in the literature while those for Model 3 were from further blood plasma experiments.

The error sum of squares was essentially the same at convergence for all four methods in Table 1 for any given model and for all data sets considered in Table 2 except in some cases where convergence did not occur in 50 iterations. The iteration cutoff was set at 50 due to computer time and cost limitations.

The number of iterations to convergence of the linear-nonlinear method was better than or equal to that for the all nonlinear method using the Hartley technique for all 25 data sets for Model 2 and for 14 data sets for Model 3 as shown in Table 2. The opposite trend was observed when the Marquardt technique was used as 9 and 10 data sets for Models 2 and 3, respectively, afforded less or equal iterations to convergence. Table 3 shows the average number of iterations to convergence for Models 2

TABLE 1

COMPARISON OF NONLINEAR AND LINEAR-NONLINEAR METHODS

USING HARTLEY AND MARQUARDT TECHNIQUES

Method	Parameter Dimensions			No. of Observations	No. of Iterations to Convergence**	Sec. per Iteration	Total Computer Time (Sec.)	Error Sum of Squares
	p	q	m	n				
				Model 1				
H-NL*			2	11	5	.40	2.00	1.8324
H-NL+L	1	1		11	3	.68	2.04	1.8324
M-NL			2	11	7	.22	1.54	1.8324
M-NL+L	1	1		11	4	.42	1.68	1.8324
				Model 2				
H-NL			3	6	6	.24	1.44	13,390.093
H-NL+L	2	1		6	3	.44	1.32	13,390.093
M-NL			3	6	6	.16	.96	13,390.093
M-NL+L	2	1		6	11	.22	2.42	13,390.093
				Model 3				
H-NL			4	20	10	.74	7.40	54,075.75
H-NL+L	2	2		20	16	1.40	22.40	54,075.77
M-NL			4	20	12	.40	4.80	54,075.77
M-NL+L	2	2		20	26	.72	18.72	54,075.79
				Model 4				
H-NL			12	80	5	2.90	14.50	21,564.5254
H-NL+L	10	2		80	3	8.34	25.02	21,564.5254
M-NL			12	80	5	1.52	7.60	21,564.5254
M-NL+L	10	2		80	3	4.20	12.60	21,564.5254

*H: Hartley L: Linear NL: Nonlinear M: Marquardt

**Convergence criteria: Relative change in the parameters = 10^{-3}; Relative change in error sum of squares = 10^{-6}

TABLE 2

COMPARISON OF THE NUMBER OF ITERATIONS TO CONVERGENCE FOR LINEAR-NONLINEAR VERSUS ALL NONLINEAR PARAMETERS FOR 25 DATA SETS

Method	Iterations to Convergence		
	Less	Same	Greater
Model 2 (25 Data Sets)			
Hartley	23	2	0
Marquardt	2	7	16
Model 3 (25 Data Sets)			
Hartley	9	5	11
Marquardt	3	7	15

TABLE 3

AVERAGE NUMBER OF ITERATIONS AND AVERAGE COMPUTER TIME PER ITERATION FOR THE NONLINEAR AND LINEAR-NONLINEAR METHODS USING HARTLEY AND MARQUARDT TECHNIQUES

Method	Average No. of Iterations to Convergence**	Average No. of Iterations to Convergence***	Average UNIVAC 1108 Computer Time Per Iteration (sec.)
	Model 2		
H-NL*	14.52	7.76(4)	.41
H-NL+L	5.76	5.76(0)	.45
M-NL	18.96	9.16(6)	.15
M-NL+L	34.88	15.64(14)	.25
	Model 3		
H-NL	23.28	12.89(7)	1.21
H-NL+L	30.08	11.69(12)	1.21
M-NL	25.88	14.53(8)	.37
M-NL+L	36.40	19.09(14)	.63

* See Table 1 for symbol definition and convergence criteria.

** All 25 data sets included in average.

*** Only those data sets where convergence occurred in less than 50 iterations included (that number given in parentheses for more than 50 iterations) and 3 using the 25 data sets for the four methods considered. There was a significant improvement in iterations using the new method employing the Hartley technique for Model 2 while the iterations were essentially the same for Model 3. However, the iterations were greater for the new method in both Models 2 and 3 using the Marquardt approach.

It was found that the method of parameter reclassification from all nonlinear to linear-nonlinear did not improve the number of iterations at convergence (except for a few isolated cases) unless the step was included during a given iteration to calculate $\underline{\beta}$ (linear parameters) for every trial value of $\underline{\alpha}$ (nonlinear parameters). This step adjusts the sum-of-squared-error to its greatest lower bound which is a function of the nonlinear parameters only. This adjustment is not made in the Hartley or Marquardt methods for the all nonlinear approach except accidentally.

Computer time for each iteration for the linear-nonlinear method compared to the all nonlinear method was essentially the same when using the Hartley approach, but was greater (approximately $1\frac{2}{3}$ times greater) for the Marquardt approach. This is emphasized in Table 3.

The model chosen to be fit, observed data, and starting parameter vector govern the final results using the described techniques so greatly that it is difficult to generalize about the merits of the methods. However, from the work accomplished thus far, one can conclude the following in regard to the new method presented:

1). The dimensionality of the vector of iterants is reduced (and thus the number of initial guesses to be made) which simplifies the sum-of-squared-error surface.

2). Because of the dimensionality reduction less sensitivity to starting guesses is achieved and, in many cases, faster convergence is obtained because of less iterations to convergence.

3). Improved results (less iterations and/or computer run time) are obtained for the linear-nonlinear method when using the Hartley technique, but not when using the Marquardt technique.

Good starting values will in most instances allow an iterative technique to converge to a solution much faster than would otherwise be possible. Also, if multiple minima exist or if there are several local minima in addition to an absolute minima, poor starting values may result in convergence to an unwanted stationary point of the sum of squares surface. The reduction in dimension of the initial parameter vector using the new method will help in this area. The starting vectors used for the four models studied in Table 1 were those supplied by the experimenter. Those for Models 2 and 3 studied in Tables 2 and 3 were obtained by the author upon analyzing the 50 data sets. The results reported could differ somewhat depending on these starting vector values, but it is highly unlikely that the conclusions would be altered.

The model chosen to be fit for a given set of observations is of prime importance in the ease of solution, regardless of the algorithm used. If the solution matrix, A or A*, is ill-conditioned then perhaps another model should be considered rather than more complicated techniques than those discussed here. Regardless, one should consider more than one model unless a great deal is known about the experiment as to its expected results and model form.

There are other areas of research which could be considered for future work based in part on this dissertation. Some of these are:

a. Consideration of a nonlinear least squares procedure which takes proper account of observational errors in both the x and y variables rather than just the y variable.

b. Incorporation of Aitken's δ^2 process (a numerical procedure discussed in most recent textbooks on numerical analysis) in the algorithm to possibly decrease the number of iterations to convergence.

c. Consideration of the "jackknife" procedure to reduce the bias of the least squares parameter estimates obtained from the algorithm presented here.

APPENDIX

COMPUTER PROGRAM FOR LEAST SQUARES ESTIMATION OF NONLINEAR PARAMETERS WHEN SOME OF THE PARAMETERS ARE LINEAR

The model for nonlinear regression for this computer program is written

$$E[Y] = F(X,A)$$

where: Y is some response variable.

E is the expectation operator.

X is the input vector controlled with NX elements.

A is a vector of parameters with M elements.

F is called the function and relates X and A to the expectation of Y.

The parameters A are in general nonlinearly involved. They can all be considered nonlinear or reclassified linear-nonlinear, depending on the method of analysis chosen. The user of this program is required to have N paired observations on X and Y ($N > M$) from which he desires a least squares estimate of A, say $\hat{A}$. The user has the option of three algorithms:

1). Marquardt's Algorithm [11].

2). Hartley's Algorithm [7].

3). A mixture of the above two algorithms.

Convergence is defined in terms of two conditions:

1). The maximum absolute relative change in the parameters less than EPS1 which is input to the program.

2). The relative change in error sum of squares (SSE) less than EPS2 which also is input to the program.

Convergence is met when both of these conditions are satisfied.

This program allows the user to fix any number of the parameters and obtain estimates of the other parameters under this constraint.

There are two primary steps to setting up a problem for this algorithm:

I. Parts of three FORTRAN subprograms must be written.

II. The data and cards specifying the options, parameters, and their initial values must be prepared.

I. Subprograms to be Written

A. Subroutine CREATE

The variables in the observations are called the input variables. As each observation is read these input variables are placed in an array U. The X and Y variables are then created from the U array. The X variables are read into the U array first followed by the Y values for any given observation. The program allows for the identification (ID) of each observation (X,Y) vector. The defined variables are as follows:

Symbol	Comment
U	Vector of input variables.
NIV	Number of input variables.
IN	Card reader designation set in subroutine SETUP (the tape, card reader, and punch unit numbers peculiar to the computer being used must be set in this subroutine by the user).
X	Vector of X variables.
NX	Number of X variables.
Y	Vector of Y variables.
ID	Identification for each (X,Y) observation vector.

Symbol	Comment
C	Vector of constants to be read in as parameters for use in this subroutine, if desired.

A format statement numbered 1 must be prepared for a given problem to read in the (ID,X,Y) values for each observation. Use G format for the (X,Y) variables and alphanumeric (not more than A8) for the ID identification. See the program listing for an example of this subroutine.

B. Function F

The specific function to be fitted is fixed in the program through this subprogram. The statements should finally produce an evaluation of the expectation of Y given the vectors X and A. This value must be designated F. In addition, there must be a subprogram within Function F designated ENTRY FSUBK(X,A,K,KEY). This function is used only when the parameters are designated linear-nonlinear, but must be included in the subroutine regardless. Of course, if it is not used it matters not what the statements of ENTRY FSUBK are. The specific function which is multiplied by each linear parameter A(K) is fixed through this subprogram. This functional value modifying each linear parameter A(K) is also designated F and the statements should produce this value. The defined variables for Function F are:

Symbol	Comment
X	Vector of X variables.
NX	Number of X variables.
A	Vector of parameter estimates.
M	Number of parameters.
C	Vector of constants to be transmitted to subroutine PD.

The defined variables for ENTRY FSUBK are:

Symbol	Comment
X	Vector of X variables.
A	Vector of parameter estimates.
K	Subscript telling which of the A parameter estimates are linear.
KEY	Integer set equal to 0 in subroutine ESTLIN - not used.

C. Subroutine PD

The estimation algorithm requires the partial derivatives of the function F with respect to the parameters. Subroutine PD fixes these derivatives in the program for each parameter. However, if estimated partial derivatives are desired (designated by an input option), then this subroutine need not contain any statements. The defined variables are:

Symbol	Comment
X	Vector of X variables.
NX	Number of X variables.
A	Vector of parameter estimates.
M	Number of parameters.
FXA	Value of the function evaluated at X and A.
C	Vector of constants defined in Function F.
P	Vector of the partial derivatives. The partial derivatives of the function with respect to each parameter are set equal to the P vector.
WATE	Factor to be used for a weighted least squares approach, if desired. The value of WATE must be designated in this subroutine. Set WATE=1.0D0 if this approach is not desired.

II. Input Data

The user of this program must specify values for certain input data. In addition, a set of control cards will be needed which direct the computer to compile and execute the program. The control cards will depend on the user's particular machine installation. The card input data format follows.

Item 1 - Information Card (Format 20A4)

One card on which the user may describe his problem. Any characters may be used. The information on this card is printed out at the beginning of the computer print out.

Item 2 - Observation Control Card (Format 7I5)

Columns	Symbol	Comment
1-5	N	Number of observations.
6-10	NIV	Number of input variables.
11-15	NX	Number of X's.
16-20	NY	Number of Y's. If NY=0, then NY is set to 1. At present, NY must be set to 0 or 1.
21-25	NC	Number of constants to be read in for use in Subroutine CREATE. If NC=0, skip item 3.
26-30	OPT1	Not used.
31-35	OT1	Data print-out option. If OT1=1, the program will not print out the original observation data.

Item 3 - Data Constants (Format 4G20.10)

This item is used to enter constants for use in the creation of the X's and Y's in Subroutine CREATE. If NC=0 in item 2, this item must be skipped.

Columns	Symbol	Comment
1-20	C(1)	First constant.
21-40	C(2)	Second constant.
.....	.....	.....
1-20	C(5)	Fifth constant.
.....	.....	Use only as many cards as necessary to enter the NC constants declared in item 2.

Item 4 - Observations (Format designated in Subroutine CREATE)

The following rules pertain to the observations:

a. Each observation must start a card record. The observations may have more than one record.

b. The first item in each observation must be the observation identification (ID). Any alphanumeric characters may be used with the maximum being 8 such characters. This identification is printed out with the original data.

c. There must be exactly as many input variables as declared in item 2. These follow the identification with the X values followed by the Y values.

Item 5 - Model Control Card (Format 3I4,4(1X,I1),5I4,10A4)

Columns	Symbol	Comment
1-4	M	Number of parameters.
5-8	NFP	Number of fixed parameters. If NFP=0, then item 7 must be deleted.
9-12	NLP	Number of linear parameters. If NLP=0, then all the parameters are considered as nonlinear; otherwise, the algorithm uses the linear-nonlinear approach for the parameter estimation.
14	OPT1	Estimated partial derivative option. If OPT1=0 then Subroutine PD is used to obtain the partial derivatives. If OPT1=1, the partial derivatives are approximated (finite differences) in Subroutine ALGOR.

16	OPT2	Algorithm option.
		OPT2=1, Marquardt Algorithm.
		OPT2=2, Hartley Algorithm.
		OPT2=3, Mixed Algorithm.

18	OPT3	Residual analysis frequency. If OPT3=0, then the residual analysis is performed only at the beginning of the iterations and after convergence. If OPT3=J($1 \leq J \leq 9$), then the analysis will be performed after every J^{th} iteration.
20	OPT4	Not used.

Columns	Symbol	Comment
21-24	MAXIT	The maximum number of iterations. If MAXIT=0, then there is no limit on the number of iterations.
25-28	KEPS1	The absolute value of the exponent in the first convergence condition. $EPS1=10^{-KEPS1}$ where the maximum absolute relative change in parameters being less than EPS1 is the first convergence condition. If KEPS1=0, then KEPS1 is set equal to 3.
29-32	KEPS2	The absolute value of the exponent in the second convergence condition. $EPS2=10^{-KEPS2}$ where the relative change in SSE being less than EPS2 is the second convergence condition. If KEPS2=0, then KEPS2 is set equal to 6.
33-36	KL	Absolute value of the exponent in LAMBDA(λ). Please see Marquardt's paper [11]. LAMBDA = 10^{-KL}. If KL=0, then KL is set equal to 2.
37-40	KN	Absolute value of the exponent in NU(ν). See Marquardt's paper [11]. $NU=10^{+KN}$. If KN=0, then KN is set equal to 1.
41-80	INFO	These columns may be used for information specific to the model. This is printed out at the beginning of each estimation procedure run.

Item 6 - List of Fixed Parameters (Format 20I4)

The indices of the parameters which are to be fixed are entered through this item. If in item 5, NFP = 0, then this item must be deleted. The list must be in ascending order.

Columns	Symbol	Comment
1-4	LFP(1)	Index of first fixed parameter.
5-8	LFP(2)	Index of second fixed parameter.
.....	.....	Continue for as many additional cards as necessary.

Item 7 - List of Linear Parameters (Format 20I4)

The indices of the parameters which are to be considered as linear are entered through this item. If in item 5, NLP=0, then this item must be deleted. The list must be in ascending order.

Columns	Symbol	Comment
1-4	LLP(1)	Index of first linear parameter.
5-8	LLP(2)	Index of second linear parameter.
.....	......	Continue for as many additional cards as necessary.

Item 8 - Parameter Initial Estimates (Format 4G20.13)

The initial estimates of the parameters are entered through this item. If NFP ≠ 0 in item 5, then the initial parameter values indexed in item 6 are considered fixed. If NLP ≠ 0 in item 6, then the initial parameter values indexed in item 7 are not used in the program as these are calculated from the nonlinear parameter initial estimates. However, initial values must be read in for <u>all</u> parameters even though they may not be used.

Columns	Symbol	Comment
1-20	A(1)	Initial estimate of first parameter.
21-40	A(2)	Initial estimate of second parameter.
.....	.....	.....
1-20	A(5)	Initial estimate of fifth parameter.
.....	.....	Use only as many cards as necessary to enter M parameters declared in item 5.

Item 9 - Data Set Termination

Item (5,6,7,8) sets may be repeated for as many different styles of analyses as desired. A blank card following item 8 will terminate the consideration of the data set. This item may be followed by another item 1, etc., for another data set. A blank card is required for the proper termination of consideration of a data set.

The following model was used to provide specific examples of the Subroutine CREATE, PD, and Function F which appear in the computer listing which follows:

$$E(Y) = A(1)EXP[A(2)X(1)] + A(3)EXP[A(4)X(1)],$$

where EXP is the exponential function.

FORTRAN V LEVEL 3, MOD 1   MAIN PROGRAM

```
      DOUBLEPRECISIONRA1,RA2,RA3,RA4,RA5,RA6,RA7,RA8,RS1,RS2,RS3,RS4,C
      COMMONRA1(465),RA2(30),RA3(30),RA4(30),RA5(30),RA6(30),RA7(30),RA8
     -(30)
     -,IA1(30),IA2(30),C(30)
     -,RS1,RS2,RS3,RS4
     -,IS1,IS2,IS3,IS4
     -,N,NX,M,IR,IP,IW,IT1,IT2,IT3
C
C     APPROPRIATE UNIT NUMBERS MUST BE SET HERE
C     IR-CARD READER, IP-PUNCH, IW-PRINTER
C     IT1 AND IT2-SCRATCH TAPES OR DISCS
C
      IR=5
      IP=-3
      IW=6
      IT1=17
      REWINDIT1
      IT2=18
      REWINDIT2
  101 CALLSETUP
      CALLALGOR
      GOTO101
      END
```

SETUP

```
      SUBROUTINESETUP
      DOUBLEPRECISIONRA1,RA2,RA3,RA4,RA5,RA6,RA7,RA8,RS1,RS2,RS3,RS4
      DOUBLEPRECISIONID,INFO,U,X,SST,SYY,SY,Y,C
      INTEGEROPT1,OT1
      DIMENSIONINFO(1),U(1),X(1)
      COMMONRA1(465),RA2(30),RA3(30),RA4(30),RA5(30),RA6(30),RA7(30),RA8
     -(30)
     -,IA1(30),IA2(30),C(30)
     -,RS1,RS2,RS3,RS4
     -,IS1,IS2,IS3,IS4
     -,N,NX,M,IR,IP,IW,IT1,IT2,IT3
      EQUIVALENCE(RA1(1),U(1)),(RA2(1),X(1)),(RA3(1),INFO(1))
     -,(RS1,SST),(RS2,SYY),(RS3,SY)
    1 FORMAT(1H153X,24HDEPARTMENT OF STATISTICS/50X,
     *29HSOUTHERN METHODIST UNIVERSITY//10X,
     *37HWEIGHTED NONLINEAR ESTIMATION PROGRAM)
    2 FORMAT(20A4/7I5)
    3 FORMAT(//11X,20A4)
    4 FORMAT(//11X,23HNUMBER OF OBSERVATIONS=I7)
    5 FORMAT(11X,26HNUMBER OF INPUT VARIABLES=I4)
    6 FORMAT(/11X,14HNUMBER OF X'S=I16)
   21 FORMAT(//11X,16HX AND Y MATRICES)
```

```
      WRITE(IW,1)
      READ(IR,2)(INFO(I),I=1,20),N,NIV,NX,NY,NC,OPT1,OT1
      WRITE(IW,3)(INFO(I),I=1,20)
      IF(OT1.EQ.1) GO TO 400
      WRITE(IW,4)N
      WRITE(IW,5)NIV
      WRITE(IW,6)NX
  400 SY=0.0
      SYY=0.0
      IF(NC.EQ.0) GO TO 300
      READ(IR,7) (C(J),J=1,NC)
    7 FORMAT(4G20.10)
  300 CONTINUE
C
C     PUT X'S AND Y ON THE DISC
C
      DO301K=1,N
      CALLCREATE(U,NIV,IR,X,NX,Y,ID,C)
      WRITE(IT1)ID,(X(J),J=1,NX),Y
      SY=SY+Y
      SYY=SYY+Y*Y
  301 CONTINUE
      REWINDIT1
      SST=SYY-SY*SY/FLOAT(N)
      IF(OT1.EQ.1) GO TO 401
      WRITE(IW,21)
      CALLPMT(U,N,NX,1,IT1,IW)
  401 RETURN
      END
```

PMT

```
      SUBROUTINEPMT(U,N,NX,NY,IT,IW)
      DOUBLEPRECISIONID,U
      INTEGERTYPE
C
C     PRINT THE X AND Y VALUES
C     TYPE 3 INDICATES X, TYPE 4 INDICATES Y
C
      DIMENSIONU(1),TYPE(40),INDEX(40)
    1 FORMAT(/11X,2HID3X,9(4X,I1,1H(I3,1H)2X))
    2 FORMAT(8X,A8,9G12.5)
      NV=NX+NY
      DO101J=1,NV
      IF(J-NX)1102,1102,102
 1102 TYPE(J)=3
      INDEX(J)=J
      GOTO101
  102 TYPE(J)=4
```

```
       INDEX(J)=J-NX
   101 CONTINUE
       NC=NV
       NCB=9
       J1=1
   201 IF(NC-NCB)701,700,700
   701 NCB=NC
   700 J2=J1+NCB-1
       WRITE(IW,1)(TYPE(J),INDEX(J),J=J1,J2)
       DO202I=1,N
       READ(IT)ID,(U(J),J=1,NV)
       WRITE(IW,2)ID,(U(J),J=J1,J2)
   202 CONTINUE
       REWINDIT
       NC=NC-NCB
       IF(NC)801,801,800
   801 RETURN
   800 J1=J2+1
       GOTO201
       END
```

ALGOR

```
       SUBROUTINEALGOR
       DOUBLEPRECISIONRA1,RA2,RA3,RA4,RA5,RA6,RA7,RA8,RS1,RS2,RS3,RS4
       DOUBLEPRECISIONLAMBDA,NU,ID,MAXRCP,INFO,X,W,A,G,V,D,P,U,T,SST,SYY
      -,SY,F,ANGLE,C,WATE,ONE,A11,A12,A13,A14,SUM
      -,EPS1,EPS2,ETA,FLTN,AY,SSE,ESTVAR,ESTSD,R1,R2,RCSSE,Y,ESTY,H,PI,DI
      -,PHI,PSI,DELTA,SSET,THETA,XI,SSE1,SSE2,DEN,CHG,RCP,DUMMY
       INTEGEROPT1,OPT2,OPT3,OPT4
       DIMENSIONX(1),W(1),A(1),T(1),P(1),D(1),U(1),G(1),V(1)
      -,LFP(1),LVP(1),LLP(30),A11(465),A12(30,30),A13(30,30),A14(465)
      -,INFO(20)
       COMMONRA1(465),RA2(30),RA3(30),RA4(30),RA5(30),RA6(30),RA7(30),RA8
      -(30)
      -,IA1(30),IA2(30),C(30)
      -,RS1,RS2,RS3,RS4
      -,IS1,IS2,IS3,IS4
      -,N,NX,M,IR,IP,IW,IT1,IT2,IT3
       EQUIVALENCE(RA1(1),W(1)),(RA2(1),X(1)),(RA3(1),A(1)),(RA4(1),T(1))
      -,(RA5(1),P(1),D(1)),(RA6(1),U(1)),(RA7(1),G(1)),(RA8(1),V(1))
      -,(RS1,SST),(RS2,SYY),(RS3,SY)
      -,(IA1(1),LFP(1)),(IA2(1),LVP(1))
     8 FORMAT(/11X,37H***ESTIMATED PARTIAL DERIVATIVES***)
    13 FORMAT(16X,2HA(I3,2H)=G18.10)
    14 FORMAT(16X,2HA(I3,2H)=G18.10,3X,5HFIXED)
    16 FORMAT(/11X,12HY STATISTICS//16X,4HSUM=G19.10/16X,5HMEAN=G18.10/16
      - X,4HSYY=G19.10/16X,4HSST=G19.10)
```

```
   37 FORMAT(13X,4G15.5,G20.10)
   61 FORMAT(/14X,19HPARAMETER ESTIMATES10X,6HCHANGE7X,10HREL CHANGE4X,
     -17HNEGATIVE GRADIENT,3X,17HCORRECTION VECTOR)
   62 FORMAT(11X,2HA(I3,2H)=G18.10,2G15.5,2G20.10)
   63 FORMAT(11X,2HA(I3,2H)=G18.10,7X,5HFIXED)
   64 FORMAT(/11X,4HSSE=G22.10,3X,14H(SST-SSE)/SST=G13.5/11X,8HMLE VAR=G
     -18.10,3X,14H(SYY-SSE)/SYY=G13.5/11X,7HMLE SD=G19.10,3X,15HREL CHAN
     -GE SSE=G12.5)
  101 ETA=1.0D-04
      EPS1=1.0D-03
      EPS2=1.0D-06
      LAMBDA=1.0D-02
      ONE=1.0D0
      NU=1.0D+01
      READ(IR,1)M,NFP,NLP,OPT1,OPT2,OPT3,OPT4,MAXIT,KEPS1,KEPS2,KL,KN,
     *(INFO(I),I=1,10)
    1 FORMAT(3I4,   4(1X,I1),5I4,10A4)
      IF(M)1001,1001,1002
 1001 RETURN
 1002 IF(KEPS1)1003,1004,1003
 1003 EPS1=10.0**(-KEPS1)
 1004 IF(KEPS2)1005,1006,1005
 1005 EPS2=10.0**(-KEPS2)
 1006 IF(KL)1007,1008,1007
 1007 LAMBDA=10.0**(-KL)
 1008 IF(KN)1009,1010,1009
 1009 NU=10.0**(+KN)
 1010 WRITE(IW,3)(INFO(I),I=1,10)
    3 FORMAT(1H155X,20HPARAMETER ESTIMATION//11X,10A4)
      WRITE(IW,5)M,EPS1,EPS2,LAMBDA,NU
    5 FORMAT(//11X,21HNUMBER OF PARAMETERS=I4//11X,11HEPSILON(1)=1PG14.2
     -/11X,11HEPSILON(2)=1PG14.2//11X,7HLAMBDA=1PG18.2/11X,3HNU=1PG22.2)
      FLTN=N
      AY=SY/FLTN
      WRITE(IW,16)SY,AY,SYY,SST
      CALL KLOCK
   67 FORMAT(80X,4HTIME,F7,2)
      IF(NFP)1021,1021,1022
C
C     READ FIXED PARAMETERS
C     READ INITIAL VALUES OF PARAMETERS TO BE ESTIMATED
C
 1022 READ(IR,11)(LFP(I),I=1,NFP)
   11 FORMAT(20I4)
 1021 IF(NLP.EQ.0)GO TO 1024
      READ(IR,11)(LLP(I),I=1,NLP)
 1024 READ(IR,2)(A(I),I=1,M)
    2 FORMAT(4G20.13)
      WRITE(IW,12)
   12 FORMAT(/11X,24HPARAMETER INITIAL VALUES/)
      LFP(NFP+1)=0
      LLP(NLP+1)=0
      J=1
      JJ-1
      K=0
      IF(NLP.EQ.0)GO TO 1025
      CALL ESTLIN(N,IT1,NX,NLP,LLP,A,A11)
 1025 CONTINUE
      DO103I=1,M
      IF(I.EQ.LFP(J))GO TO 102
      IF(I.EQ.LLP(JJ))GO TO 1103
 1102 K=K+1
      LVP(K)=I
      WRITE(IW,13)I,A(I)
      GOTO103
 1103 JJ=JJ+1
      WRITE(IW,15)I,A(I)
   15 FORMAT(16X,2HA(,I3,2H)=,G18.10,3X,6HLINEAR)
      GO TO 1029
  102 J=J+1
      WRITE(IW,14)I,A(I)
 1029 T(I)=A(I)
      G(I)=0.0
      V(I)=0.0
  103 CONTINUE
      NVP=K
      LVP(NVP+1)=0
      NE=NVP*(NVP+1)/2
      CALLRESID(X,N,NX,IT1,A,M,IW,SSE,0,C)
      ESTVAR=SSE/FLTN
      ESTSD=DSQRT(ESTVAR)
      R1=(SST-SSE)/SST
      R2=(SYY-SSE)/SYY
      RCSSE=R1
      WRITE(IW,64)SSE,R1,ESTVAR,R2,ESTSD,RCSSE
      IF(OPT1-1)1031,1032,1031
 1032 WRITE(IW,8)
 1031 L=0
  301 L=L+1
      CALL KLOCK(TIME)
      WRITE(IW,67) TIME
      CALL KLOCK
      WRITE(IW,21)L
   21 FORMAT(///11X,13H*** ITERATIONI5,4H ***//20X,6HLAMBDA9X,5HDELTA10X
     -,5HTHETA11X,3HPHI15X,3HSSE)
      IJ=0
      DO302I=1,NVP
      U(I)=0
      DO302J=1,I
      IJ=IJ+1
      W(IJ)=0.0
  302 CONTINUE
```

```
      IF(NLP.EQ.0) GO TO 314
      DO 8102 I=1,NVP
      DO 8101 J=1,NLP
 8101 A12(I,J)=0.0D0
 8102 CONTINUE
  314 DO 321 K=1,N
      READ(IT1)ID,(X(J),J=1,NX),Y
      ESTY=F(X,NX,A,M,C)
      IF(OPT1-1)1311,311,1311
C     ANALYTIC PARTIAL DERIVATIVES
 1311 CALLPD(X,NX,A,M,ESTY,C,P,WATE)
      GO TO 313
C    ESTIMATED PARTIAL DERIVATIVES
  311 DO316I=1,NVP
      IVP=LVP(I)
      H=ETA*DABS(T(IVP))
      IF(H)7654,7654,7655
 7654 H=ETA
 7655 T(IVP)=T(IVP)+H
      P(IVP)=(F(X,NX,T,M,C)-ESTY)/H
      T(IVP)=A(IVP)
  316 CONTINUE
  313 IF(NLP.EQ.0)GO TO 312
      DO 8200 I=1,NVP
      IVP=LVP(I)
      DO 8199 J=1,NLP
      JLP=LLP(J)
 8199 A12(I,J)=A12(I,J)+P(IVP)*P(JLP)
 8200 CONTINUE
  312 IJ=0
      DO322I=1,NVP
      IVP=LVP(I)
      PI=P(IVP)
      U(I)=U(I)+(Y-ESTY)*PI*WATE
      DO322J=1,I
      JVP=LVP(J)
      IJ=IJ+1
      W(IJ)=W(IJ)+PI*P(JVP)*WATE
  322 CONTINUE
  321 CONTINUE
      IF(NLP.EQ.0)GO TO 324
C     MULT A11 TIMES A12 TRANS. TO GIVE A13
      DO 20 J=1,NVP
      DO 19 I=1,NLP
      SUM=0.0D0
      DO 18 K=1,I
      KK=I*(I-1)/2+K
   18 SUM=SUM+A11(KK)*A12(J,K)
      IF(I.EQ.NLP)GO TO 4
      KMP=I+1
      DO 17 K=KMP,NLP
      KKK=K*(K-1)/2+I
   17 SUM=SUM+A11(KKK)*A12(J,K)
    4 A13(I,J)=SUM
   19 CONTINUE
   20 CONTINUE
C     MULT A12 TIMES A13 TO GIVE A14
      DO 30 J=1,NVP
      DO 29 I=1,J
      SUM=0.0D0
      DO 28 K=1,NLP
      KK=J*(J-1)/2+I
      SUM=SUM+A12(J,K)*A13(K,I)
   28 A14(KK)=SUM
   29 CONTINUE
   30 CONTINUE
      IJ=0
      DO 323 I=1,NVP
      DO 323 J=1,I
      IJ=IJ+1
      W(IJ)=W(IJ)-(A14(IJ)*WATE)
  323 CONTINUE
  324 REWIND IT1
C
C     PRECONDITION THE SYSTEM OF EQUS.
      II=0
      IJ=0
      DO331I=1,NVP
      II=II+1
      DI=ONE/DSQRT(W(II))
      W(II)=ONE
      D(I)=DI
      IVP=LVP(I)
      G(IVP)=U(I)
      U(I)=U(I)*DI
      DO332J=1,I
      IJ=IJ+1
      IF(I-J)1332,332,1332
 1332 W(IJ)=W(IJ)*D(J)*DI
  332 CONTINUE
  331 CONTINUE
      IF(LAMBDA) 999,501,999
  999 WRITE(IT2)(W(IJ),IJ=1,NE),(U(I),I=1,NVP)
      REWINDIT2
      IF(OPT2-2)1501,501,1501
C    MARQUARDT ALGORITHM
 1501 PHI=1.0
      PSI=LAMBDA/NU
  401 READ(IT2)(W(IJ),IJ=1,NE),(U(I),I=1,NVP)
      REWINDIT2
      II=0
      DO402I=1,NVP
      II=II+I
      W(II)=1.0+PSI
  402 CONTINUE
      CALLCF1(W,NVP,DELTA)
      CALLCF4(W,NVP,U)
 9000 DO403I=1,NVP
      IVP=LVP(I)
      V(IVP)=U(I)*D(I)
      T(IVP)=A(IVP)+V(IVP)
  403 CONTINUE
      IF(NLP.EQ.0) GO TO 9010
      CALL ESTLIN(N,IT1,NX,NLP,LLP,T,A11)
 9010 CALLRESID(X,N,NX,IT1,T,M,IW,SSET,1,C)
      IF(NVP-1)9003,9002,9003
 9002 THETA=0.0D0
      GO TO 9004
 9003 THETA=ANGLE(V,G,M)
 9004 WRITE(IW,37)PSI,DELTA,THETA,PHI,SSET
      IF(SSET-SSE)411,1411,1411
 1411 PSI=NU*PSI
      IF(DABS((SSET-SSE)/SSE)-EPS2) 500,500,401
  ~~411~~ IF(OPT2-1)1541,541,1541  ←  411  LAMBDA = PSI
 1541 XI=1.0
      SSE1=SSET
      GOTO512
  500 READ(IT2)(W(IJ),IJ=1,NE),(U(I),I=1,NVP)
      REWINDIT2
      WRITE(IW,555)
  555 FORMAT(18H ABANDON MARQUARDT)
C     HARTLEY'S ALGORITHM
  501 IF(NVP.EQ.1) GO TO 9001
      CALLCF1(W,NVP,DELTA)
      CALLCF4(W,NVP,U)
 9001 DO505I=1,NVP
      IVP=LVP(I)
      V(IVP)=U(I)*D(I)
  505 CONTINUE
      IF(NVP-1)9006,9005,9006
 9005 THETA=0.0D0
      GO TO 9007
 9006 THETA=ANGLE(V,G,M)
 9007 LAMBDA=0.0
      XI=1.0
  503 CALLHART(A,V,NVP,LVP,XI,T)
      IF(NLP.EQ.0)GO TO 9008
      CALL ESTLIN(N,IT1,NX,NLP,LLP,T,A11)
 9008 CALLRESID(X,N,NX,IT1,T,M,IW,SSE1,1,C)
      WRITE(IW,37)LAMBDA,DELTA,THETA,XI,SSE1
      IF(XI-1.0)511,512,511
  512 XI=0.5*XI
      SSE2=SSE1
      IF(DABS(XI)-EPS2) 525,525,503
 511 DEN=SSE-2.0*SSE1+SSE2
     IF(DEN)1521,1521,521
1521 IF(SSE2-SSE)532,512,512
 521 PHI=0.5*XI(3.0*SSE-4.0*SSE1+SSE2)/DEN
     CALLHART(A,V,NVP,LVP,PHI,T)
     IF(NLP.EQ.0)GO TO 9009
     CALL ESTLIN(N,IT1,NX,NLP,LLP,T,A11)
9009 CALLRESID(X,N,NX,IT1,T,M,IW,SSET,1,C)
     WRITE(IW,37)LAMBDA,DELTA,THETA,PHI,SSET
     IF(SSET-SSE)1524,524,524
1524 IF(SSET-SSE1)1525,525,525
1525 IF(SSET-SSE2)541,532,532
 524 IF(SSE1-SSE)525,512,512
 525 IF(SSE2-SSE1)532,1532,1532
1532 SSET=SSE1
     PHI=XI
     GOTO537
 532 SSET=SSE2
     PHI=2.0*XI
 537 CALLHART(A,V,NVP,LVP,PHI,T)
 541 WRITE(IW,61)
     IF(NLP.EQ.0)GO TO 545
     CALL ESTLIN(N,IT1,NX,NLP,LLP,T,A11)
 545 CONTINUE
     MAXRCP=0.0
     J=1
     JJ=1
     DO601I=1,M
     IF(I.EQ.LFP(J))GO TO 1602
     IF(I.EQ.LLP(JJ))GO TO 1603
     GO TO 602
1602 WRITE(IW,63)I,A(I)
     J=J+1
     GOTO601
1603 A(I)=T(I)
     WRITE(IW,1604)I,A(I)
1604 FORMAT(11X,2HA(,I3,2H)=,G18.10,7X,6HLINEAR)
     JJ=JJ+1
     GO TO 601
 602 CHG=A(I)-T(I)
     IF(A(I))1632,632,1632
1632 RCP=CHG/A(I)
     GOTO631
 632 RCP=1.0E+35
 631 A(I)=T(I)
     WRITE(IW,62)I,A(I),CHG,RCP,G(I),V(I)
     RCP=DABS(RCP)
     IF(RCP-MAXRCP)2001,2001,2002
2002 MAXRCP=RCP
2001 CONTINUE
 601 CONTINUE
```

```
      RCSSE=(SSE-SSET)/SSE
      SSE=SSET
      ESTVAR=SSE/FLTN
      ESTSD=DSQRT(ESTVAR)
      R1=(SST-SSE)/SST
      R2=(SYY-SSE)/SYY
      WRITE(IW,64)SSE,R1,ESTVAR,R2,ESTSD,RCSSE
      IF(MAXRCP-EPS1)1691,691,691
 1691 IF(RCSSE-EPS2)2691,691,691
 2691 WRITE(IW,66)
   66 FORMAT(///11X,35H***** CONVERGENCE IS MANIFEST *****//)
      CALL KLOCK(TIME)
      WRITE(IW,67) TIME
      CALLRESID(X,N,NX,IT1,A,M,IW,DUMMY,0,C)
      GOTO101
  691 IF(OPT3)1692,692,1692
 1692 IF(FLOAT(L)/FLOAT(OPT3)-FLOAT(L/OPT3))692,3001,692
 3001 CALLRESID(X,N,NX,IT1,A,M,IW,DUMMY,0,C)
  692 IF(L-MAXIT)301,695,301
  695 WRITE(IW,2)(A(I),I=1,M)
      GOTO101
      END
```

ESTLIN

```
      SUBROUTINE ESTLIN(N,IT1,NX,NLP,LLP,A,A11)
      DOUBLE PRECISION W,U,ID,X,A,D,DELTA,DI,FI,FJ,Y,FSUBK,A11
      COMMON W(465),X(30)
      DIMENSION U(30),LLP(1),D(30),A(1),A11(1)
      IJ=0
      DO 100 I=1,NLP
      U(I)=0.0
      DO 100 J=1,I
      IJ=IJ+1
      W(IJ)=0.0
  100 CONTINUE
      DO 300 NN=1,N
      READ (IT1)ID,(X(J),J=1,NX),Y
      IJ=0
      KEY=0
      DO 200 I=1,NLP
      ILP=LLP(I)
      FI=FSUBK(X,A,ILP,KEY)
      U(I)=U(I)+Y*FI
      DO 200 J=1,I
      IJ=IJ+1
      JLP=LLP(J)
      FJ=FSUBK(X,A,JLP,KEY)
      W(IJ)=W(IJ)+FI*FJ
  200 CONTINUE
  300 CONTINUE
      REWIND IT1
C
C     PRECONDITION THE SYSTEM OF EQUATIONS
      II=0
      IJ=0
      DO 400 I=1,NLP
      II=II+1
      DI=1.0D0/DSQRT(W(II))
      W(II)=1.0D0
      A11(II)=W(II)
      D(I)=DI
      U(I)=U(I)*DI
      DO 390 J=1,I
      IJ=IJ+1
      IF(I.EQ.J)GO TO 390
      W(IJ)=W(IJ)*D(J)*DI
      A11(IJ)=W(IJ)
  390 CONTINUE
  400 CONTINUE
C
C     SOLVE THE SYSTEM
      IF(NLP.EQ.1) GO TO 700
      CALL CF1(W,NLP,DELTA)
      CALL CF4(W,NLP,U)
      CALL CF3(A11,NLP)
  700 II=0
      IJ=0
      DO 600 I=1,NLP
      II=II+1
      A11(II)=A11(II)*D(I)*D(I)
      DO 590 J=1,I
      IJ=IJ+1
      IF(I.EQ.J)GO TO 590
      A11(IJ)=A11(IJ)*D(J)*D(I)
  590 CONTINUE
  600 CONTINUE
      DO 500 I=1,NLP
      ILP=LLP(I)
  500 A(ILP)=U(I)*D(I)
      RETURN
      END
```

RESID

```
      SUBROUTINERESID(X,N,NX,IT,A,M,IW,SSE,K,C)
      DOUBLEPRECISIONID,A,X,SSE,U,V,Y,FLTN,Z,R,ESTY,BETA1,BETA2,SKEW,C,F
      DIMENSIONA(1),X(1) ,C(1)
     -,U(4),V(4)
   62 FORMAT(/11X,17HRESIDUAL ANALYSIS//20X,2HID11X,1HY8X,13HESTIMATE OF
    - Y5X,8HRESIDUAL)
  63 FORMAT(16X,A8,3G15.5)
  65 FORMAT(//17X,6HMOMENT5X,10HABOUT ZERO10X,10HABOUT MEAN)
  66 FORMAT(16X,I5,2G20.10)
  67 FORMAT(//11X,8HBETA(1)=G18.10,3X,8HBETA(2)=G18.10,3X,9HSKEWNESS=G1
    -7.10)
     IF(K-1)1606,606,1606
1606 WRITE(IW,62)
     DO601I=1,4
     U(I)=0.0
 601 CONTINUE
     DO602L=1,N
     READ(IT)ID,(X(J),J=1,NX),Y
     ESTY=F(X,NX,A,M,C)
     R=Y-ESTY
     WRITE(IW,63)ID,Y,ESTY,R
     Z=R
     DO605I=1,4
     U(I)=U(I)+Z
     Z=Z*R
 605 CONTINUE
 602 CONTINUE
     REWINDIT
     SSE=U(2)
     FLTN=N
     DO603I=1,4
     U(I)=U(I)/FLTN
 603 CONTINUE
     V(1)=0.0
     V(2)=U(2)-U(1)**2
     V(3)=U(3)-3.0*U(1)*U(2)+2.0*U(1)**3
     V(4)=U(4)-4.0*U(1)*U(3)+6.0*(U(1)**2)*U(2)-3.0*U(1)**4
     WRITE(IW,65)
     WRITE(IW,66)(I,U(I),V(I),I=1,4)
     BETA1=V(3)**2/(V(2)**3)
     BETA2=V(4)/(V(2)**2)
     SKEW=DSQRT(BETA1)*(BETA2+3.0)/(2.0*(5.0*BETA2-6.0*BETA1-9.0))
     WRITE(IW,67)BETA1,BETA2,SKEW
     RETURN
 606 SSE=0.0
     DO607L=1,N
     READ(IT)ID,(X(J),J=1,NX),Y
     SSE=SSE+(Y-F(X,NX,A,M,C))**2
 607 CONTINUE
     REWINDIT
     RETURN
     END
```

CF1

```
     SUBROUTINECF1(A,N,DELTA)
     DOUBLEPRECISIONA,DELTA,S,AII
```

```
      DIMENSIONA(1)
      DELTA=1.0
      I=N
      NN=N*(N+1)/2
      IJ=NN
  101 J=I
      II=IJ
  102 K=I
      KI=II
      KJ=IJ
      S=0.0
  104 K=K+1
      IJ(K-N)1301,1301,103
 1301 KI=KI+K-1
      KJ=KJ+K-1
      S=S+A(KI)*A(KJ)
      GOTO104
  103 IF(J-I)801,802,803
  801 A(IJ)=(A(IJ)-S)/AII
      GOTO803
  802 AII=DSQRT(A(II)-S)
  803 IJ=IJ-1
      J=J-1
      IF(J)1102,1102,102
 1102 DELTA=DELTA*AII*AII
      A(II)=AII
      I=I-1
      IF(I)1101,1101,101
 1101 RETURN
      END
```

CF3

```
      SUBROUTINECF3(A,N)
      DOUBLE PRECISION A,T,S,TII
      DIMENSIONA(1),T(40)
      IJ=0
      DO401I=1,N
      IM1=I-1
      L=IJ
      IF(IM1.EQ.0)GOTO407
      DO402J=1,IM1
      L=L+1
      T(J)=A(L)
  402 CONTINUE
  407 TII=A(L+1)
      KO=0
      DO401J=1,I
      S=0.0
      IF(IM1.EQ.0)GOTO408
      DO404K=1,IM1
      IF(K.LE.J)KJ=KO+K
      IF(K.GT.J)KJ=KJ+K-1
      S=S+T(K)*A(KJ)
  404 CONTINUE
      S=-S/TII
  408 IF(J.EQ.I)S=1.0/(TII**2)+S
      IJ=IJ+1
      A(IJ)=S
      KO=KO+J
  401 CONTINUE
      RETURN
      END
```

CF4

```
      SUBROUTINECF4(T,N,X)
      DOUBLEPRECISIONT,X,S
      DIMENSIONT(1),X(1)
C (1) SOLUTION OF T'Y=B
      II=N*(N+1)/2
      I=N
      X(I)=X(I)/T(II)
  201 II=II-I
      IJ=II
      IP1=I
      I=I-1
      S=X(I)
      DO202J=IP1,N
      IJ=IJ+J-1
      S=S-T(IJ)*X(J)
  202 CONTINUE
      X(I)=S/T(II)
      IF(I-1)1201,1201,201
C (2) SOLUTION OF TX=Y
 1201 IJ=1
      X(1)=X(1)/T(1)
      DO301I=2,N
      S=X(I)
      IM1=I-1
      IJ=IJ+1
      DO302J=1,IM1
      S=S-T(IJ)*X(J)
      IJ=IJ+1
  302 CONTINUE
      X(I)=S/T(IJ)
  301 CONTINUE
      RETURN
      END
```

HART

```
      SUBROUTINEHART(A,V,NVP,LVP,PHI,T)
      DOUBLEPRECISIONA,V,T,PHI
      DIMENSIONA(1),V(1),T(1),LVP(1)
      DO101I=1,NVP
      IVP=LVP(I)
      T(IVP)=A(IVP)+PHI*V(IVP)
  101 CONTINUE
      RETURN
      END
```

ANGLE

```
      DOUBLEPRECISIONFUNCTIONANGLE(X,Y,N)
      DOUBLEPRECISIONX,Y,SXX,SYY,SXY,XI,YI,COSINE
      DIMENSIONX(1),Y(1)
      SXX=0.0
      SYY=0.0
      SXY=0.0
      DO101I=1,N
      XI=X(I)
      YI=Y(I)
      SXX=SXX+XI*XI
      SYY=SYY+YI*YI
      SXY=SXY+XI*YI
  101 CONTINUE
      COSINE=SXY/DSQRT(SXX*SYY)
      ANGLE=DACOS(COSINE) *57.295779D0
      RETURN
      END
```

CREATE

```
      SUBROUTINECREATE(U,NIV,IN,X,NX,Y,ID,C)
      DOUBLEPRECISIONID,U,X,Y,C,A
      DIMENSIONU(1),X(1),C(1)
C***  FORMAT  (1) NUMBER IT 1
C***          (2) FIRST FIELD A-FIELD OF MAX WIDTH 8 FOR ID
C***          (3) FOLLOW WITH G- OR D-FIELDS FOR INPUT VARIABLES
C*****  PLACE FORMAT AFTER THIS CARD  ***************************************
C****************************************************************************
C   THE FOLLOWING PROGRAM CALCULATES THE X-MATRIX USING 'CREATE'
    1 FORMAT(A5,2G10.0)
C*****  PLACE FORMAT BEFORE THIS CARD  **************************************
      READ(IN,1)ID,(U(J),J=1,NIV)
C*****  PLACE FORTRAN STATEMENTS CREATING X AND Y VARIABLES AFTER THIS CARD**
      X(1)=U(1)
      Y=U(2)
```

```
C***********************************************************************
C*****  PLACE FORTRAN STATEMENTS CREATING X AND Y VARIABLES BEFORE THIS CARD
      RETURN
      END
```

F

```
      DOUBLEPRECISIONFUNCTIONF(X,NX,A,M,C)
      DOUBLEPRECISIONX,A,C,SUMAX,E1,E23
      DIMENSIONA(1),X(1),C(1)
C*****  PLACE FORTRAN STATEMENTS FOR EVALUATING F AFTER THIS CARD  ********
C***********************************************************************
      E1=A(2)*X(1)
      IF(E1.GT.170.D0) E1=170.D0
      E23=A(4)*X(1)
      IF(E23.GT.170.D0)E23=170.D0
      C(1)=DEXP(E1)
      C(2)=DEXP(E23)
      F=A(1)*C(1) + A(3)*C(2)
      RETURN
      ENTRY FSUBK(X,A,K,KEY)
      E1=A(K+1)*X(1)
      IF(E1.GT.170.D0) E1=170.D0
      F=DEXP(E1)
      RETURN
C***********************************************************************
C*****  PLACE FORTRAN STATEMENTS FOR EVALUATING F BEFORE THIS CARD  ******
      END
```

PD

```
      SUBROUTINEPD(X,NX,A,M,FXA,C,P,WATE)
      DOUBLEPRECISIONX,A,FXA,P,C,WATE
      DOUBLEPRECISIONU,V
      DIMENSIONX(1),A(1),P(1),C(1)
C*****  PLACE FORTRAN STATEMENTS FOR EVALUATING PARTIAL DERIV AFTER THIS
     -CARD ***
C***********************************************************************
C   CALCULATES FIRST PARTIALS
      WATE=1.0D0
      P(1)=C(1)
      P(2)=X(1)*A(1)*C(1)
      P(3)=C(2)
      P(4)=X(1)*A(3)*C(2)
C***********************************************************************
C***********************************************************************
C***********************************************************************
```

```
C*****  PLACE FORTRAN STATEMENTS FOR EVALUATING PARTIAL DERIV BEFORE THIS
     -CARD
      RETURN
      END
```

LIST OF REFERENCES

[1] Beale, E. M. L. (1960). "Confidence Regions in Non-Linear Estimation." Journal Royal Statistical Society, Series B, 22, 41-76.

[2] Drane, J. W. (1967). "Difference Equations and Maximum Likelihood Estimates in Nonlinear Regression." Ph.D. Dissertation 67-14,334, University Microfilms Inc., Ann Arbor, Michigan, 74.

[3] Drane, J. W. and Schucany, W. R. (1969). "Another Approach to Nonlinear Least Squares." Presented at the 7th Annual Symposium on Biomathematics and Computer Science in the Life Sciences at Houston, Texas, March 24-26, 1969.

[4] Draper, N. R. and Smith, H. (1966). Applied Regression Analysis, John Wiley and Sons, Inc., New York, 273-274.

[5] Flanagan, P. D. and Vitale, P.A., and Mendelsohn, J. (1969). "A Numerical Investigation of Several One-Dimensional Search Procedures in Nonlinear Regression Problems." Technometrics, 11, 265-284.

[6] Guttman, I. and Meeter, D. A. (1965). "On Beale's Measure of Non-Linearity." Technometrics, 7, 623-637.

[7] Hartley, H. O. (1961). "The Modified Gause-Newton Method for the Fitting of Non-Linear Regression Functions by Least Squares." Technometrics, 3, 269-280.

[8] Hartley, H. O. (1964). "Exact Confidence Regions for the Parameters in Non-Linear Regression Laws." Biometrika, 51, 347-353.

[9] Jennrich, R. I. and Sampson, P. F. (1968). "Application of Stepwise Regression to Non-Linear Estimation." Technometrics, 10, 63-72.

[10] Lawton, W. H. and Sylvestre, E. A. (1970). "Elimination of Linear Parameters in Nonlinear Regression." Presented at the 130th Annual Meeting of ASA, Biometric Society, ENAR and WNAR at Detroit, Michigan, Dec. 27-30, 1970.

[11] Marquardt, D. W. (1963). "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." Journal of Society Indust. Appl. Math., 11, 431-441.

[12] Marquardt, D. W. (1970). "Generalized Inverses, Ridge Regression, Biased Linear Estimation, and Nonlinear Estimation." *Technometrics*, *12*, 591-612.

[13] Marquardt, D. W. (1970). "Generalized Inverses in Nonlinear Estimation - Theory and Example." Presented at the 130th Annual Meeting of ASA, Biometric Society, ENAR and WNAR at Detroit, Michigan, Dec. 27-30, 1970.

[14] Meeter, D. A. (1966). "On A Theorem Used In Nonlinear Least Squares." *Journal SIAM Applied Math.*, *14*, 1176-1179.

[15] Nelson, D. L. and Lewis, T. O. (1970). "A Method for the Solution of Non-Linear Least Squares Problems When Some of the Parameters Are Linear." *The Texas Journal of Science*, *21*, 480.

[16] Papaioannou, T. and Kempthorne, O. (1970). "Parallel Tangents and Steepest Descent Optimization Algorithm - A Computer Implementation with Application to Partially Linear Models." *ARL 70-0117* Iowa State University, Dept. of Statistics, Ames, Iowa.

[17] Richards, F. S. G. (1961). "A Method of Maximum-likelihood Estimation." *Journal Royal Statistical Society*, *Series B*, *23*, 469-475.

[18] Ross, G. J. S. (1970). "The Efficient Use of Function Minimization in Non-linear Maximum-likelihood Estimation." *Applied Statistics*, *19*, 205.

[19] Seibert, G.B. Jr. (1970). "Estimation and Confidence Intervals for Quantal Response or Sensitivity Data." *Ph.D. Dissertation* presented to Southern Methodist University Faculty.

[20] Spang, H. A. III (1962). "A Review of Minimization Techniques for Nonlinear Functions." *SIAM Review*, *4*, 343-365.

[21] Spillman, W. J. (1933). "Use of the Exponential Yield Curve in Fertilizer Experiments." *Technical Bulletin No. 348*, U.S. Dept. of Agriculture, Washington, D.C.

[22] Stevens, W. L. (1951). "Asymptotic Regression." *Biometrics*, *7*, 247-267.

[23] Walling, D. (1968). "Non-Linear Least Squares Curve Fitting When Some Parameters Are Linear." *The Texas Journal of Science*, *20*, 119-124.

[24] Williams, E. J. (1962). "Exact Fiducial Limits in Non-linear Estimation." *Journal Royal Statistical Society*, *Series B*, *24*, 125-139.