A STUDY OF THREE ALGORITHMS FOR NONLINEAR LEAST SQUARES PARAMETER ESTIMATION

by

MICKEY LINN STILSON

B.S., Kansas State University, 1971

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Statistics

KANSAS STATE UNIVERSITY

1976

Approved by:

George A. Milliken

Major Professor

Table of Contents

Section Page

1. Introduction..1

2. Formulation of model and technique...............................4

2.1 General technique...4

2.1.1 Taylor series technique (Gauss-Newton procedure).........5

2.1.2 Method of steepest descent................................7

2.2 Hartley's modified Gauss-Newton procedure......................11

2.3 Marquardt's procedure..12

2.4 Spiral algorithm...16

2.5 Distributional properties of parameter estimates...............20

3. Some examples...22

4. Constrained parameter estimation................................32

4.1 An example using growth curves.................................34

References...37

Appendix...38

A user's guide for NONLIN......................................38

NONLIN...41

Acknowledgements...64

1. Introduction.

Scientists are frequently interested in investigating the relationship between some response or dependent variable, denoted by y, and a vector of independent variables, denoted by $\underline{x}$. Assume the relationship is postulated to be some function f involving a vector of parameters $\underline{\theta}$, as

$$y=f(\underline{x},\underline{\theta}).$$

For an experimental situation where y is a random variable, the assumed relationship is

$$E(y)=f(\underline{x},\underline{\theta})$$

and the model with additive error structure can be expressed as

$$y=f(\underline{x},\underline{\theta})+\varepsilon.$$

Thus over the course of n observations, we construct the model

$$\underline{y}=\underline{f}(\underline{x},\underline{\theta})+\underline{\varepsilon},$$

where $\underline{y}$, $\underline{f}(\underline{x},\underline{\theta})$, and $\underline{\varepsilon}$ are n×1 vectors.

In order to more accurately classify the type of models to be analyzed, we present the following definitions ([9]):

Definition 1.1 A model is $y=f(\underline{x};\underline{\theta};\underline{\varepsilon})$, where y is the value of an observed random variable which is to measure the phenomena under study (dependent variable), $\underline{x}$ is a vector of constants or other observed random variables (independent variables), $\underline{\theta}$ is a vector of unknown parameters, $\underline{\varepsilon}$ is a vector of unobserved random variables with some assumed distribution, and $f(\cdot,\cdot,\cdot)$ is some known mathematical function of $\underline{x}$, $\underline{\theta}$, and $\underline{\varepsilon}$.

Definition 1.2 A model is defined to be a linear model if $y=f(\underline{x},\underline{\theta})+h(\underline{\varepsilon})$, where $f(\underline{x},\underline{\theta})$ is a linear function of the elements of $\underline{\theta}$.

Note the additive error structure implied in the above definition of a linear model. An example is $y=\theta_0+\theta_1x+\theta_2x^2+\varepsilon$.

Definition 1.3 A model is defined to be essentially linear if there exists a transformation $\ell(y)=\ell(f(\underline{x},\underline{\theta},\underline{\varepsilon}))$ such that $\ell(y)=g(\underline{x},\underline{\theta})+h(\underline{\varepsilon})$, where $g(\underline{x},\underline{\theta})$ is a linear function of $\underline{\theta}$, and $h(\underline{\varepsilon})$ is a function of $\underline{\varepsilon}$ only. An example of an essentially linear model is

$$y=(e^{\theta_0}e^{\theta_1x})(\varepsilon)$$

as $ln(y)=\theta_0+\theta_1x+ln(\varepsilon)$. The model $y=\exp(\theta_0+\theta_1x)+\varepsilon$ is nonlinear because we cannot make a transformation to a linear model retaining an additive error structure. The additive error structure is important in order to apply least squares to making interval estimates about $\underline{\theta}$ and examining various distributional properties.

Definition 1.4 Any model $y=f(\underline{x},\underline{\theta},\underline{\varepsilon})$ that is not linear or essentially linear is defined to be nonlinear.

Statistical theory offers many techniques for obtaining estimators of $\underline{\theta}$ from the model $y=f(\underline{x},\underline{\theta})+\underline{\varepsilon}$, including maximum likelihood, Bayesian, and least squares. For an account of general methods of obtaining estimators see [10](Chapter VII). For most techniques some objective function of $\underline{\theta}$, say $\Phi(\underline{\theta})$, is to be optimized. Examples of Φ include risk functions (Bayesian estimation), likelihood functions (maximum likelihood estimation), and sums of squares (least squares estimation). In this paper we restrict ourselves to obtaining least squares estimators for $\underline{\theta}$ from the model $y=f(\underline{x},\underline{\theta})+\underline{\varepsilon}$, where f is generally a nonlinear function in $\underline{\theta}$. These estimators are also maximum likelihood estimators when normality is assumed. Since linear estimation is a special case of nonlinear estimation, all results discussed will apply equally to linear and nonlinear estimation.

In our study we will begin with a general formulation of a technique of finding a least squares estimator $\hat{\underline{\theta}}$ for $\underline{\theta}$ in $y=f(\underline{x},\underline{\theta})+\underline{\varepsilon}$. The technique utilizes a Taylor series linear approximation to f and develops an iterative scheme to approach $\hat{\underline{\theta}}$. The scheme is generally referred to as the Gauss-Newton or Taylor series method. We then study three modifications to the general technique of the Gauss-Newton method. These include the modified Gauss-Newton [6], the Marquardt [8], and the Spiral [7] algorithms. These modifications are based on the premise that a procedure that converges in fewer iterations and/or with less computational effort is an improvement.

Section 3 is devoted to several examples and the problem of parameter estimation under constraints is considered in Section 4. The appendix documents a computer program developed by the author incorporating the algorithms of the modified Gauss-Newton, Marquardt, and Spiral techniques. The results of Section 4 are included in the program so that constrained estimation is possible using either the modified Gauss-Newton or Spiral algorithms. The appendix includes a user's guide to the program along with sample output.

In this study all theorems are quoted without proof and often with less than complete rigor. The reader is referred to the references for detail.

2. Formulation of model and technique.

2.1 General technique.

Consider the model

$$y=f(x_1,x_2,\ldots,x_k;\theta_1,\theta_2,\ldots,\theta_p)+\varepsilon.$$

By letting $\underline{x}=(x_1,\ldots,x_k)'$ and $\underline{\theta}=(\theta_1,\ldots,\theta_p)'$, the above model can be expressed as

$$y=f(\underline{x},\underline{\theta})+\varepsilon. \tag{2.1}$$

If there are n observations of the form $y_i,x_{1i},x_{2i},\ldots,x_{ki}$ for $i=1,\ldots,n$, the above model can be written as

$$y_i=f(x_{1i},\ldots,x_{ki};\theta_1,\ldots,\theta_p)+\varepsilon_i,$$

or, following the notation of (2.1),

$$y_i=f(\underline{x}_i,\underline{\theta})+\varepsilon_i. \tag{2.2}$$

Letting $\underline{\varepsilon}=(\varepsilon_1,\ldots,\varepsilon_n)'$, we make the usual assumptions that $E(\underline{\varepsilon})=\underline{0}$ and $E(\underline{\varepsilon\varepsilon'})=\sigma^2\underline{I}$, i.e., that the errors are identically distributed with zero means, equal variances, σ^2, and zero covariances. For the purpose of obtaining confidence intervals and testing hypotheses, we will later assume $\underline{\varepsilon}$ is normally distributed as $\underline{\varepsilon}\sim N(\underline{0},\sigma^2\underline{I})$, but this assumption is not necessary for estimation purposes.

The least squares procedure involves determining a value of $\underline{\theta}$ which minimizes the sum of squared deviations of the observations from their expected value. The objective function can be expressed as

$$\Phi(\underline{\theta})=\sum_{i=1}^{n}\{y_i-f(\underline{x}_i,\underline{\theta})\}^2. \tag{2.3}$$

Since y_i and $\underline{x}_i$ are observations, only $\underline{\theta}$ is an unknown variable in $\Phi(\underline{\theta})$.

We define the least squares estimator of $\underline{\theta}$, denoted by $\hat{\underline{\theta}}$, as that value which minimizes $\Phi(\underline{\theta})$. From [4] we note that under the assumption $\underline{\varepsilon}\sim N(\underline{0},\sigma^2\underline{I})$, $\hat{\underline{\theta}}$ can also be shown to be the maximum likelihood estimator of $\underline{\theta}$.

To determine a suitable value for $\hat{\underline{\theta}}$, we differentiate (2.3) with respect to $\underline{\theta}$, providing the p normal equations in $\hat{\underline{\theta}}$, a solution,

$$\sum_{i=1}^{n}\{ y_i-f(\underline{x}_i,\hat{\underline{\theta}})\}\left[\frac{\partial f(\underline{x}_i,\underline{\theta})}{\partial\theta_j}\right]_{\underline{\theta}=\hat{\underline{\theta}}} =0$$

for j=1,...,p, which are to be solved for $\hat{\underline{\theta}}$. This is generally not an easy task as direct solutions are not available and an iterative process must be used. Not only are the equations difficult to solve, but frequently multiple solutions exist ([4]).

2.1.1 Taylor series technique (Gauss-Newton procedure).

Suppose we have a preliminary estimate of $\underline{\theta}$, denoted by

$$\underline{\theta}^0=(\theta_1^0,\theta_2^0,\dots,\theta_p^0) ,$$

obtained from previous experience or knowledge, or as the result of intelligent guessing. For $\underline{\theta}$ sufficiently close to $\underline{\theta}^0$, expansion of $f(\underline{x}_i,\underline{\theta})$ in a Taylor series about $\underline{\theta}^0$, keeping only first order terms, yields

$$(2.4)\qquad f(\underline{x}_i,\underline{\theta})\doteq f(\underline{x}_i,\underline{\theta}^0)+\sum_{j=1}^{p}\left[\frac{\partial f(\underline{x}_i,\underline{\theta})}{\partial\theta_j}\right]_{\underline{\theta}=\underline{\theta}^0}(\theta_j-\theta_j^0).$$

Letting $f_i^0=f(\underline{x}_i,\underline{\theta}^0)$, $\delta_j^0=(\theta_j-\theta_j^0)$, and $Z_{ij}^0=\left[\frac{\partial f(\underline{x}_i,\underline{\theta})}{\partial\theta_j}\right]_{\underline{\theta}=\underline{\theta}^0}$,

the first order approximation of the model in (2.2) becomes

(2.5) $$y_i-f_i^0 \doteq \sum_{j=1}^{p} \delta_j^0 Z_{ij}^0+\varepsilon_i \; ; \quad i=1,\ldots,n.$$

Writing $$\underline{Z}^0 = \begin{bmatrix} Z_{11}^0 \cdots & Z_{1p}^0 \\ \vdots & \\ Z_{n1}^0 \cdots & Z_{np}^0 \end{bmatrix},$$

$\underline{y}=(y_1,\ldots,y_n)'$, $\underline{\delta}^0=(\delta_1^0,\ldots,\delta_n^0)'$, and $\underline{f}^0=(f_1^0,\ldots,f_n^0)'$,

a matrix form of (2.5) is

(2.6) $$(\underline{y}-\underline{f}^0)=\underline{Z}^0\underline{\delta}^0+\underline{\varepsilon},$$

which is a model linear in the unknown parameters $\underline{\delta}^0$. The normal equations for (2.6) are

(2.7) $$\underline{Z}^{0\prime}\underline{Z}^0\underline{\delta}^0=\underline{Z}^{0\prime}(\underline{y}-\underline{f}^0)$$

which, on assuming $\underline{Z}^0$ is of full rank, have solutions

(2.8) $$\hat{\underline{\delta}}^0=(\underline{Z}^{0\prime}\underline{Z}^0)^{-1}\underline{Z}^{0\prime}(\underline{y}-\underline{f}^0) \; .$$

In using (2.4), we have approximated the nonlinear function f by a plane in the region of $\hat{\underline{\theta}}$. Letting $\underline{\theta}^1=\underline{\theta}^0+\hat{\delta}^0$, the vector $\hat{\underline{\delta}}^0$ can be thought of as a correction vector giving a new estimate, $\underline{\theta}^1$, of $\hat{\underline{\theta}}$, which is the best estimate obtainable under the linear approximation. If $\Phi(\underline{\theta}^1)=0$, then $\underline{\theta}^1=\hat{\underline{\theta}}$ and we have a solution. In all likelihood this is not the case, so we now let $\underline{\theta}^1$ be our new estimate, or initial value, and repeat the procedure, deriving a new correction vector and hence a new estimate $\underline{\theta}^2$. In general

(2.9) $$\underline{\theta}^{j+1}=\underline{\theta}^j+\hat{\underline{\delta}}^j=\underline{\theta}^j+(\underline{Z}^{j\prime}\underline{Z}^j)^{-1}\underline{Z}^{j\prime}(\underline{y}-\underline{f}^j),$$

with obvious notation.

We continue the process until convergence is reached. Following [8] the process will be considered to have converged upon obtaining a correction vector $\underline{\delta}^j$ such that $\dfrac{|\delta_i^j|}{\tau+|\theta_i^j|}<\varepsilon$, for i=1,...,p, and some small $\varepsilon>0$, say 10^{-5}, and some small τ, say 10^{-3}. The presence of τ is to allow the test under the possibility $\hat{\theta}_i=0$ for some i.

For the special case where $y_i=f(\underline{x}_i,\underline{\theta})+\varepsilon_i$ is a linear model, $\underline{\delta}^1=\underline{0}$, i.e. the process converges in one step (although most programs written for the procedure will calculate $\underline{\overset{1}{\delta}}\doteq\underline{0}$ anyway).

From [4],referring to the general process described above,

> " The linearization procedure has possible draw-backs for some problems in that
> 1. It may converge very slowly; that is, a very large number of iterations may be required before the solution stabilizes even though the sum of squares...may decrease consistently as j increases. This sort of behavior is not common but can occur.
> 2. It may oscillate widely, continuously reversing direction, and often increasing, as well as decreasing the sum of squares. Nevertheless the solution may stabilize eventually.
> 3. It may not converge at all, and even diverge, so that the sum of squares increases iteration after iteration without bound."

Despite these serious draw-backs, the technique, commonly referred to as the Gauss-Newton procedure, is useful, and will work sucessfully on a wide variety of nonlinear problems ([4]).

To circumvent the problem of wide oscillation and divergence, we discuss three proposed modifications to the general technique. But first the method of steepest descent is discussed as two of the modifications partially incorporate it into their modifications. As in the above discussion, the development of the method of steepest descent is from [4]. A more general and theoretical development is found in [1].

2.1.2 Method of steepest descent.

From some initial guess, $\underline{\theta}^j$, it is desired to seek an iterative value $\underline{\theta}^{j+1}$, such that $\Phi(\underline{\theta}^{j+1})<\Phi(\underline{\theta}^j)$ in a sequential manner that leads to convergence. From elementary calculus we know that the direction in which $\Phi(\underline{\theta})$ decreases lies along the vector

$$(2.10) \qquad \underline{\delta}^{*}=\left[\left(-\frac{\partial\Phi(\underline{\theta})}{\partial\theta_1},-\frac{\partial\Phi(\underline{\theta})}{\partial\theta_2},\ldots,-\frac{\partial\Phi(\underline{\theta})}{\partial\theta_p}\right)\right]_{\underline{\theta}=\underline{\theta}^j} .$$

Thus if $\underline{\delta}^j=\rho\underline{\delta}^*$ for $0<\rho\leqq 1$, then for some ρ, $\Phi(\underline{\theta}^j+\underline{\delta}^j)\leqq\Phi(\underline{\theta}^j)$. Hence at the j^{th} step, $\underline{\delta}^*$ is searched by varying ρ until we find a point at which $\Phi(\underline{\theta}^j+\underline{\delta}^*)\leqq\Phi(\underline{\theta}^j)$.

By this process we avoid the possibility of divergence or wild oscillation mentioned in the discussion of the Taylor series technique. But often the method of steepest descent converges very slowly, so slowly, in fact, as to make the method unadvisable as a sole technique. The problem occurs for models whose sums of squares surface consists of a long, narrow trough in the vicinity of the minimum. An example of this is the model

$$(2.11) \qquad f(x_i)=\theta_1+e^{\theta_2 x_i}+\varepsilon_i \quad \text{for } x_i>0,$$

since $\frac{\partial f}{\partial\theta_1}=1$ and $\frac{\partial f}{\partial\theta_2}=x_i\exp(\theta_2 x_i)$ which can be very large. Thus a small change in θ_2 will produce a much larger change in the sums of squares function than will a similar change in θ_1.

To illustrate the differences between the Taylor series and steepest descent vectors produced, consider the function

$$(2.12) \qquad y_i=5+e^{.5x_i} \quad \text{where } x_1=1,\ x_2=2, \text{ and } x_3=3.$$

Figure 2.1 graphs $\Phi(\underline{\theta})$ for various values of $\underline{\theta}$ in the vicinity of the true values $\theta_1=5$, $\theta_2=.5$. Suppose, not knowing $\underline{\theta}$, we supply as initial

ILLEGIBLE

THE FOLLOWING DOCUMENT (S) IS ILLEGIBLE DUE TO THE PRINTING ON THE ORIGINAL BEING CUT OFF

ILLEGIBLE

$\theta_2 \backslash \theta_1$	4.0	4.2	4.4	4.6	4.8	5.0	5.2	5.4	5.6	5.8
.1	2.569E1	2.254E1	1.963E1	1.696E1	1.453E1	1.234E1	1.040E1	8.687E0	7.218E0	5.989E0
.2	2.039E1	1.758E1	1.502E1	1.269E1	1.061E1	8.760E0	7.155E0	5.789E0	4.664E0	3.779E0
.3	1.442E1	1.205E1	9.922E0	8.035E0	6.388E0	4.981E0	3.814E0	2.888E0	2.201E0	1.754E0
.4	8.239E0	6.435E0	4.870E0	3.546E0	2.461E0	1.617E0	1.012E0	6.477E¯1	5.232E¯1	6.387E¯1
.5	3.000E0	1.320E0	1.080E0	4.800E¯1	1.200E¯1	4.930E¯32	1.200A¯1	4.800E¯1	1.080E0	1.320E0
.6	1.164E0	1.022E0	1.119E0	1.456E0	2.033E0	2.851E0	3.908E0	5.205E0	6.743E0	8.520E0
.7	7.723E0	8.738E0	1.011E1	1.167E1	1.340E1	1.550E1	1.777E1	2.029E1	2.304E1	2.603E1
.8	3.241E1	3.507E1	3.797E1	4.112E1	4.450E1	4.812E1	5.198E1	5.608E1	6.042E1	6.500E1
.9	9.379E1	9.853E1	1.035E2	1.087E2	1.142E2	1.199E2	1.258E2	1.320E2	1.384E2	1.451E2
1.0	2.268E2	2.342E2	2.419E2	2.498E2	2.580E2	2.664E2	2.751E2	2.840E2	2.931E2	3.025E2

Sums of Squares Surface for Model (2.12)

Figure 2.1

guesses $\theta_1^0=4.2$, $\theta_2^0=.8$. As can be seen in Figure 2.1, $\Phi(\underline{\theta}^0)=35.07$.

As presented in section 2.1.1, we form the matrix of partials,

$$\underline{Z}=\begin{bmatrix} \dfrac{\partial f(x_1,\underline{\theta})}{\partial\theta_1} & \dfrac{\partial f(x_1,\underline{\theta})}{\partial\theta_2} \\ \dfrac{\partial f(x_2,\underline{\theta})}{\partial\theta_1} & \dfrac{\partial f(x_2,\underline{\theta})}{\partial\theta_2} \\ \dfrac{\partial f(x_3,\underline{\theta})}{\partial\theta_1} & \dfrac{\partial f(x_3,\underline{\theta})}{\partial\theta_2} \end{bmatrix}$$

or

$$\underline{Z}=\begin{bmatrix} 1 & x_1e^{\theta_2x_1} \\ 1 & x_2e^{\theta_2x_2} \\ 1 & x_3e^{\theta_2x_3} \end{bmatrix},$$

and the vector $\underline{f}^0=\begin{bmatrix} f(x_1,\underline{\theta}^0) \\ f(x_2,\underline{\theta}^0) \\ f(x_3,\underline{\theta}^0) \end{bmatrix}$.

From the given data, we compute $\underline{y}=\begin{bmatrix} 6.6487 \\ 7.7183 \\ 9.4817 \end{bmatrix}$.

Thus $\underline{\theta}^1=\underline{\theta}^0+(\underline{Z}'\underline{Z})^{-1}\underline{Z}'(\underline{y}-\underline{f}^0)=(4.77,\ 0.608)'$. $\Phi(\underline{\theta}^1)=2.407$, and so by the Taylor series techinque we have found a new point $\underline{\theta}^1$ such that $\Phi(\underline{\theta}^1)<\Phi(\underline{\theta}^0)$, and hence we have a successful iteration.

Following the method of steepest descent,

$$\underline{\delta}^*=(-6.95,\ -203.47)'.$$

This correction is too severe to allow graphing in Figure 2.1, so we use $\rho=.001$, and compute $\underline{\delta}^1=\rho\underline{\delta}^*=(-.00695,\ -.20347)'$. Thus $\underline{\theta}^1=\underline{\theta}^0+\underline{\delta}^1=(4.19,\ .5965)'$, and $\Phi(\underline{\theta}^1)=0.9496$.

Our two updates serve to illustrate fairly universal properties of the two techniques: the update $\underline{\theta}^1$ by steepest descent produces a greater reduction in the objective function, but the $\underline{\theta}^1$ of the Taylor series method is "closer" to $\underline{\theta}$ in terms of units of θ_1 and θ_2. One should note how the steepest descent corrections will begin a hemstitch pattern across the sums of squares trough depicted in Figure 2.1 as it continually moves across the valley in the direction of steepest slope.

When $\underline{\theta}^0$, the initial guess, is "far enough" away from $\hat{\underline{\theta}}$, steepest descent updates are more satisfactory than corrections produced by the Taylor series method, but as $\underline{\theta}$ approaches $\hat{\underline{\theta}}$, Newtonian steps are more satisfactory, as illustrated above.

2.2 Hartley's modified Gauss-Newton procedure.

Intuition tells us that, given "enough" time (i.e., iterations), the method of steepest descent will ultimately lead us to a minimum in $\Phi(\underline{\theta})$, albiet a spurious, local minimum is always a possibility. However, this is not the case with the Gauss-Newton procedure. If the linear approximation of $\Phi(\underline{\theta})$ in the vicinity of $\underline{\theta}^j$ is a particularly poor one, the method can easily lead to divergence, due mainly to an inappropriate step size rather than an inappropriate direction. An obvious "solution" exists; go some portion of the distance of the Gauss-Newton correction vector, as, for some $\underline{\delta}^j$ correction obtained by the Gauss-Newton method, there exists ρ, $0 \leq \rho \leq 1$, such that $\Phi(\underline{\theta}^j + \underline{\delta}^j) \leq \Phi(\underline{\theta}^i)$. The proof of existence and subsequent convergence of the procedure is given in [6]. Hartley shows that if $\underline{\theta}^0$ is within a bounded convex set S of the parameter space spanned by $\underline{\theta}$, and if, for

$$Q = \liminf_{\bar{S}} \Phi(\underline{x}, \underline{\theta}),$$

where $\bar{S}$ is the complement of S, there exists a $\underline{\theta}^*$ in the interior of S such that

$$\Phi(\underline{x},\underline{\theta}^*)<0,$$

then the process as modified above will converge (to at least a local minimum, if $\underline{\theta}^0$ is not properly chosen).

In application Hartley suggests that $\Phi(\underline{\theta}^j+\rho\underline{\delta}^j)$ be evaluated at $\rho=0$, $\rho=\frac{1}{2}$, and $\rho=1$. The three values describe a parabola in ρ, from which a minimum is obtained from

$$\rho_{min}=\tfrac{1}{2}+\tfrac{1}{4}(\Phi(\rho=0)-\Phi(\rho=1)\ )/\ (\Phi(\rho=1)-2\Phi(\rho=\tfrac{1}{2})+\Phi(\rho=0)\ )$$

If

$$\Phi(\underline{\theta}^j+\rho_{min}\underline{\delta}^j)>\Phi(\underline{\theta}^j),$$

then the computations are repeated using $\frac{1}{2}\underline{\delta}^j$, and continued until an appropriate ρ_{min} is found.

2.3 Marquardt's procedure.

As previously discussed, Newtonian steps are to be preferred to those in the direction of steepest descent. Marquardt and others ([8]) have examined a number of nonlinear problems and found that typically the correction vectors produced by a Gauss-Newton method and the direction of steepest descent are 80-90^0 apart. This is seen also in our earlier example (see Figure 2.1). But a small enough step in the direction of steepest descent will always produce a reduction in the objective function, $\Phi(\underline{\theta})$, whereas no suitable correction may be found along the vector given by the Taylor series approximation (at least not along the part searched by most computer programs). Hence it seems desirable that as we reach a point in our search for $\hat{\underline{\theta}}$ where a Newtonian step does not produce a suitable step, we would use corrections given by steepest descent until Newtonian steps can be resumed. Marquardt ([8]) suggests an algorithm that interpolates between the correction vectors of the Gauss-Newton procedure and steepest descent. The theoretical basis for the algorithm is as follows: At the j^{th} iteration, let

$$\underline{A}=(\underline{Z}^{j})'\underline{Z}^{j} \quad \text{and} \quad \underline{\nu}=(\underline{y}-\underline{f}^{j}).$$

Then the Gauss-Newton correction vector, which we will now denote by $\underline{\delta}_t$, is given by the solution of

$$(2.3.1) \qquad \underline{A}\underline{\delta}_t=(\underline{Z}^{j})'\underline{\nu}$$

and the direction of steepest descent lies along the vector $\underline{\delta}_g$, where

$$(2.3.2) \qquad \underline{\delta}_g=(\underline{Z}^{j})'\underline{\nu}.$$

The following three theorems are due to Marquardt (see [8] for proofs).

Theorem 1. Let $\lambda \geqq 0$ be arbitrary and let $\underline{\delta}$ satisfy the equation

$$(2.3.3) \qquad (\underline{A}+\lambda\underline{I})\underline{\delta}=(\underline{Z}'^{j})\underline{\nu},$$

where $\underline{I}$ is the identity matrix of size p×p, p being the number of parameters to be estimated. Then $\underline{\delta}$ minimizes $\phi(\underline{\theta})$ on the sphere whose radius $||\underline{\delta}_r||$ satisfies $||\underline{\delta}_r||^2= ||\underline{\delta}||^2$.

Theorem 2. Let $\underline{\delta}(\lambda)$ be the solution of (2.3.3) for a given λ. Then $||\underline{\delta}(\lambda)||^2$ is a continuous decreasing function of λ, such that as $\lambda\to\infty$, $||\underline{\delta}(\lambda)||^2\to 0$.

Theorem 3. Let γ be the angle between $\underline{\delta}$ and $\underline{\delta}_g$. Then γ is a continuous monotone decreasing function of λ such that as $\lambda\to\infty$, $\gamma\to 0$. Since $\underline{\delta}_g$ is independent of λ, it follows that $\underline{\delta}$ rotates toward $\underline{\delta}_g$ as $\lambda\to\infty$.

Examining (2.3.3) we see that for $\lambda=0$ (2.3.3) is equivalent to (2.3.1), that is, $\underline{\delta}=\underline{\delta}_t$, and we take a Newtonian step. As λ becomes large (2.3.3) can be approximated by (2.3.2) (that is, λ dominates the maximum characteristic root of $\underline{A}$), so that for large λ we move approximately in the direction of steepest descent. For $\lambda=0$ the step size is that produced by the Gauss-Newton procedure. Theorem 2 tells us that as λ increases, the step size decreases asymptotically to zero, so that we will always obtain a suitable step $\underline{\delta}^j$ such that

$$\Phi(\underline{\theta}^j+\underline{\delta}^j)\leq\Phi(\underline{\theta}^j)$$

for an appropriate choice of λ, and hence convergence is guaranteed (even if only to a local minimum).

One additional step is necessary before we have a viable procedure. Adding λ to each diagonal element of $\underline{A}$ (as in (2.3.3)) will not be useful if the elements are of widely varying orders of magnitude. To circumvent the problem of needing to add a different λ to each diagonal element, Marquardt suggests rescaling the $\underline{A}$ matrix in terms of the standard deviations of the first partials. This amounts to a rescaling of the parameter space. As an added effect, we succeed in stabilizing $\underline{A}$ for the purpose of inversion. Letting $\underline{g} = \underline{Z}'\underline{\nu}$, define $\underline{A}^*$ and $\underline{g}^*$ by

$$\underline{A}^*=(a^*_{ij})=\left(\frac{a_{ij}}{\sqrt{a_{ii}a_{jj}}}\right) \text{ and}$$

$$\underline{g}^*=(g^*_j)=\left(\frac{g_j}{\sqrt{a_{jj}}}\right).$$

Solving for the Gauss-Newton correction $\underline{\delta}^*_t$, we use

$$\underline{A}^*\underline{\delta}^*_t=\underline{g}^*,$$

and

$$\underline{\delta}_t=(\delta_j)=\left(\frac{\delta^*_j}{\sqrt{a_{jj}}}\right).$$

The algorithm is now clear. At the j^{th} step (iteration) construct

(2.3.4) $$(\underline{A}^{*j}+\lambda^{j}\underline{I})\underline{\delta}^{*j}=\underline{g}^{*j}$$

and solve for $$\underline{\delta}^{j}=(\delta_i^{j})=(\frac{\delta_i^{*j}}{\sqrt{a_{ii}^{j}}}).$$

As before, $$\underline{\theta}^{j+1}=\underline{\theta}^{j}+\underline{\delta}^{j},$$

if we have selected λ^{j} so that

(2.3.5) $$\Phi(\underline{\theta}^{j+1})\leq\Phi(\underline{\theta}^{j}).$$

It is clear from the theorems that a (large) λ always exists such that (2.3.5) is satisfied, unless $\underline{\theta}^{j}$ is already at $\hat{\underline{\theta}}$, the minimum (global or local, as the case may be) of $\Phi(\underline{\theta})$. But constant use of a large λ will produce results similar to the steepest descent technique. Thus, when the sums of squares surface, $\Phi(\underline{\theta})$, is reasonably well approximated by linearization, we wish to use a small λ so that we take Newtonián steps, and use a large λ for gradient steps only as necessary for convergence. Accordingly, Marquardt gives the following strategy:

Let $\upsilon>1$.

Let λ^{j-1} denote the value of λ from the previous iteration (where $\lambda^{0}=.01$, say, initially). Then the trial values for λ^{j} are λ^{j-1}/υ and λ^{j-1}.

(i) Compute $\Phi(\underline{\theta}^{j+1},\lambda^{j-1}/\upsilon)$ and $\Phi(\underline{\theta}^{j+1},\lambda^{j-1})$. If $\Phi(\underline{\theta}^{j+1},\lambda^{j-1}/\upsilon)\leq\Phi(\underline{\theta}^{j})$, then $\lambda^{j}=\lambda^{j-1}/\upsilon$.

(ii) If $\Phi(\underline{\theta}^{j+1},\lambda^{j-1}/\upsilon)>\Phi(\underline{\theta}^{j})$ and $\Phi(\underline{\theta}^{j+1},\lambda^{j-1})\leq\Phi(\underline{\theta}^{j})$, then $\lambda^{j}=\lambda^{j-1}$.

(iii) Otherwise, increase λ successively by multiplying by υ until for the smallest integer ω, $\Phi(\underline{\theta}^{j+1},\lambda^{j-1}\upsilon^{\omega})\leq\Phi(\underline{\theta}^{j})$; then use $\lambda^{j}=\lambda^{j-1}\upsilon^{\omega}$.

From [8]:

> Typically, condition (iii) is met only rarely. Thus it is most often required that (2.3.4) be solved for two values of

> λ^j at each iteration. One such solution is required for the standard Taylor series method. The extra linear equation solution is generally much less computational effort than the evaluation of the $\underline{A}^*$ matrix, so that the small proportional increase in computation per iteration is more than offset by the gain in the power of an iteration.

To which we add: if the number of parameters is fairly "small" in relation to the number of observations.

2.4 Spiral algorithm.

Jones, [7], agrees that Newtonian steps are preferable and that steepest descent corrections should only be used as necessary, but he sees a major disadvantage in Marquardt's procedure in that a matrix inversion is required to generate each search point. Thus, he proposes an algorithm which searches roughly the same area as Marquardt's (this deserves further comment later in the development), but which generates search points between Newtonian and steepest descent corrections by vector addition rather than by matrix inversion. Though not specifically mentioned as a goal, we will note that his proposal also returns to a true Newtonian step immediately, rather than waiting a few iterations until the operation λ^j/υ reduces λ sufficiently. This feature probably accounts for much of the improved performance Jones claims over Marquardt's procedure (on problems for which Newtonian steps are successful for a large majority of iterations).

The essence of the Spiral algorithm Jones proposes is as follows:

In agreement with Marquardt's procedure, $\underline{\theta}^{j+1}$ always exists in the plane formed by the Taylor series correction vector and the line of steepest descent such that

$$\Phi(\underline{\theta}^{j+1}) \leqq \Phi(\underline{\theta}^{j}).$$

Figure 2.2, drawn in this defined plane, shows the point O, which is $\underline{\theta}^j$; the vector $\overline{OT}$, which is the Taylor series correction vector, and the vector $\overline{OD}$, where D is chosen along the path of steepest descent such that $||\overline{OT}||=||\overline{OD}||$.

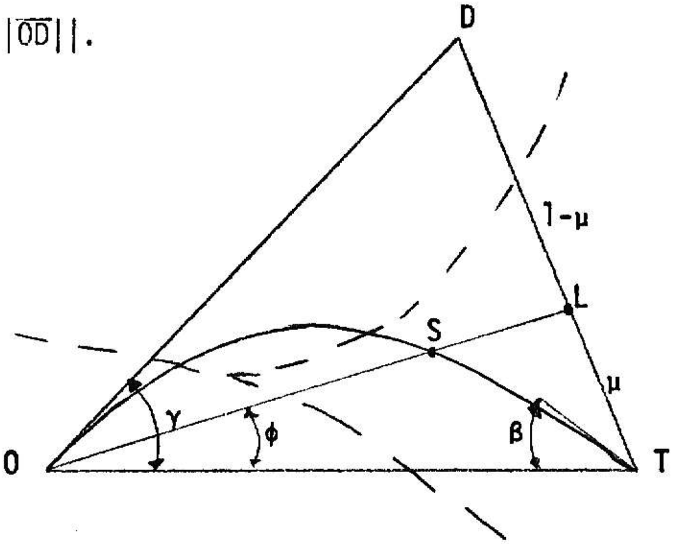

Figure 2.2

As we wish to progress as much as possible at each iteration and keep computations at a minimum, strategy demands that $\underline{\theta}^{j+1}$ be as far from $\underline{\theta}^j$ as possible and the number of evaluations of $\Phi(\underline{\theta})$ (and hence $f(\underline{x},\underline{\theta})$) be kept to a minimum. Hence $\underline{\theta}^{j+1}$ will be chosen as the first point such that $\Phi(\underline{\theta}^{j+1}) \leqq \Phi(\underline{\theta}^{j})$.

It follows that the point T in Figure 2.2 is the first to be investigated. If this does not produce a reduction in $\Phi(\underline{\theta})$, then a linear approximation of $\Phi(\underline{\theta})$ does not extend well to T from O. Hence $\Phi(\underline{\theta})$ has some minimum "trough" that curves in one of the two directions shown by the hatched curves in Figure 2.2. At this point Jones claims "Since the overall strategy tends to give base points [point O, that is, $\underline{\theta}^j$] on the outside shoulder of the valley, it is reasonable to assume that the valley is moving away from the line $\overline{OT}$." Thus he suggests the search be conducted next along the spiral OTS, as to be described shortly, in order to try to intercept the "valley" (trough). The example discussed in section 2.1.2 can be shown to be an exception to this idea. Hence we suggest that the line $\overline{OT}$ be searched as in the modified Gauss-Newton procedure of section 2.2 before attempting a search elsewhere. This strategy is incorporated in the computer program described in the appendix.

When no suitable correction is found at T (or along $\overline{OT}$), we must search the area between $\overline{OT}$ and $\overline{OD}$, as in Marquardt's procedure. As with Marquardt's procedure, we need to approach $\overline{OD}$ in such a fashion that the step size approaches zero, so that convergence can again be guaranteed. Jones suggests a search along the spiral OTS and has found the most suitable spiral to be (expressed in polar coordinates, see Figure 2.2) given by

$$r=r_0(1-\phi\cos\beta-(1-\gamma\cos\beta)(\phi/\gamma)^2)$$

where r is the distance OS, and r_0 is the distance OT. The sequence of points S to be investigated are defined from a sequence of points L generated on TD in the ratio $\mu:(1-\mu)$ (see Figure 2.2). Jones suggests that successive values of μ be computed from

$$\mu_{n+1}=2\mu_n/(1+\mu_n)$$

so that the points $\{L_n\}$ become closer together as they approach D on TD.

The coordinates of L, (ξ,ϕ), can be derived as

$$\tan\phi=\frac{\mu\sin\gamma}{1-\mu+\mu\cos\gamma},$$

and

$$\xi=\frac{r_0\mu\sin\gamma}{\sin\phi}.$$

If O is the origin, the coordinates $\underline{s}$ of S are given by the coordinates $\underline{t}$ of T and $\underline{d}$ of D by

$$\underline{s}=\underset{\xi}{\underline{r}}\{\mu\underline{d}+(1-\mu)\underline{t}\}.$$

As shown, each of the successive search points S is generated by a weighted sum of two vectors, as opposed to the matrix inversion required in Marquardt's procedure. As previously suggested, we stop as soon as we find a point S such that

$$\Phi(\underline{\theta}^j+\underline{s})\leq\Phi(\underline{\theta}^j).$$

One should see that since the spiral is searched at only a discrete set of points, it is possible to jump from one side of the trough to the other. Jones suggests that an "interpolation be performed whenever three consecutive sums of squares along the spiral are convex downwards, the sum of squares being regarded as a function of μ."

Work with this algorithm has led us to suggest two refinements, in addition to the one already mentioned. First, it is a distinct possibility that the trough of the surface $\Phi(\underline{\theta})$ curves through OT in Figure 2.2, contrary to Jones' statement. If the angle γ is fairly large, the search along the spiral OTS will often lead to points $\{S_n\}$ far away from the trough. Machine limitations on magnitudes of numbers enter into play, as $\Phi(\underline{\theta}^j+\underline{s})$ can become very large, overflowing many machines. Hence the computer program in the appendix suspends any overflow messages produced by the IBM 370 FORTRAN IV compiler, and sets $\Phi(\underline{\theta}^j+\underline{s})$ to a predetermined constant (10^{74}). Secondly, if for two consecutive search points s_n and s_{n+1} we have

$$\Phi(\underline{\theta}^j+\underline{s}_{n+1})\geq\Phi(\underline{\theta}^j+\underline{s}_n)\geq\Phi(\underline{\theta}^j),$$

then the angle β (see Figure 2.2) is cut in half and a new spiral is searched that now lies closer to the line OT (β is initially set to half of γ, as suggested by Jones). These two suggestions are not theoretical faults of the algorithm, but are necessary for implementation on computers with finite computational capabilities.

2.5 Distributional properties of parameter estimates.

For the model

$$y_i=f(\underline{x}_i,\underline{\theta})+\varepsilon_i,$$

we assume that the ε_i's are independent, identically, normally distributed with mean zero and variances σ^2, i.e.,

$$\underline{\varepsilon}\sim N(\underline{0},\sigma^2\underline{I}).$$

Hence $$E(y_i)=f(\underline{x}_i,\underline{\theta}).$$

The density function for an observation y_i is given by

$$g_i(y_i,\underline{\theta})=(2\pi\sigma^2)^{-\frac{1}{2}}\exp\{-\frac{1}{2\sigma^2}\;(y_i-f(\underline{x}_i,\underline{\theta}))^2\}\;.$$

Thus, the likelihood function for $\underline{\theta}$, $LH(\underline{\theta})$, is (for a sample of size n)

$$LH(\underline{\theta})=(2\pi\sigma^2)^{-\frac{n}{2}}\exp\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-f(\underline{x}_i,\underline{\theta})\;)^2\},$$

and the log of the likelihood function, written $L(\underline{\theta})$, is

$$L(\underline{\theta})=-\frac{n}{2}\ln(2\pi)-\frac{n}{2}\ln\sigma^2-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\;(y_i-f(\underline{x}_i,\underline{\theta}))^2$$

$$=-\frac{n}{2}\ln(2\pi)-\frac{n}{2}\ln\sigma^2-\frac{1}{2\sigma^2}\,\phi(\underline{\theta}).$$

It is now apparent that to maximize the likelihood function by suitable choice of $\underline{\theta}$, we can minimize $\Phi(\underline{\theta})$ by choice of $\underline{\theta}$. Thus the least squares estimate of $\underline{\theta}$ is also the maximum likelihood estimate for $\underline{\theta}$.

Since $\hat{\underline{\theta}}$ is a maximum likelihood estimate for $\underline{\theta}$, a central limit theorem gives us that asymptotically,

$$\hat{\underline{\theta}} \sim MVN(\underline{\theta},\underline{V}),$$

where

$$\underline{V} = \sigma^2(\underline{Z}'\underline{Z})^{-1}.$$

(Note that $(\underline{Z}'\underline{Z})^{-1}$ is evaluated at $\underline{\theta}$, the true, but unknown, parameter value.) It can be shown that $\hat{\underline{\theta}}$ is a consistent estimator of $\underline{\theta}$ and thus $\underline{Z}'\underline{Z}|_{\underline{\theta}=\hat{\underline{\theta}}}$ is a consistent estimator of $\underline{Z}'\underline{Z}|_{\theta}$. It follows that

$$\hat{\sigma}^2 = \frac{1}{n-p} \sum_{i=1}^{n} (y_i - f(\underline{x}_i,\underline{\theta}))^2$$

is a consistent estimator of σ^2, and that

$$\hat{\underline{V}} = \hat{\sigma}^2(\underline{Z}'\underline{Z}|_{\underline{\theta}=\hat{\underline{\theta}}})^{-1}$$

is a consistent estimator of $\underline{V}$. A $(1-\alpha)100\%$ asymptotic approximate confidence interval about $\underline{a}'\underline{\theta}$ for a given vector $\underline{a}$ is

$$\underline{a}'\hat{\underline{\theta}} \pm t_{\alpha/2}(n-p)\sqrt{\underline{a}'\hat{\underline{V}}\underline{a}}\ .$$

The computer program in the appendix provides the residual sums of squares, from which $\hat{\sigma}^2$ can be obtained, and $(\underline{Z}'\underline{Z})^{-1}$ evaluated at the last iterative estimate for $\hat{\underline{\theta}}$, so that confidence intervals involving $\underline{\theta}$ can be formed.

3. Some examples.

Four examples of application of the three algorithms developed in the previous section are now presented. The algorithms are incorporated in a computer program, NONLIN, written by the author and documented in the appendix along with a user's guide. It is important to notice that none of the algorithms are completely successful in that performance is related to starting values and the type of nonlinear function.

Initial parameter estimates provide the most critical problem; all of the theory for the procedures is dependent on starting the iterative process within the sphere of convergence for the least squares solution. Hence a successful estimator is often not obtained until after much searching for satisfactory starting values, if at all. Reasonable values can often be selected from a knowledge of the process, previous work, or, in the case of a few parameters and a relatively simple function, by using a few data points and solving for the unknown parameters.

The first example comes from animal science and involves estimating the parameters for a simple exponential growth curve, modeling weight of cows versus time. The model is

$$y_i=\theta_1-\theta_2\exp(-\theta_3x_i)+\varepsilon_i \tag{3.1}$$

where y_i is the weight of the animal at time x_i (time measured to the nearest month after birth). Data for a particular animal is given in Table 3.1. After 18 months a cyclical pattern dominates the data due to yearly calving. The model makes no attempt to account for this, fitting, instead, a curve through something akin to an average weight through the cycle.

Table 3.1

Data for Growth Curve Model, Eq. (3.1)

Month*	Weight (lbs)
0	64
2	100
3	130
4	160
5	205
6	270
7	305
8	310
9	310
10	346
11	315
12	375
13	380
14	415
15	450
16	535
17	550
19	540
20	660
21	730
22	700
23	650
24	670
25	610
26	470
27	610
28	615
29	675
29	700
31	715
32	710
33	690
34	650
34	670
36	640
37	630
38	540
39	610
40	670
41	720
42	745
42	775
44	775
44	780
46	745
47	680
48	690
48	665
50	645
51	635
52	690
53	690
54	750
55	770
56	820
56	825
58	780
59	855
60	830
61	800
62	640
63	770
64	810
65	875
66	905
66	935

* Month after birth.

Starting values for this example present no particular problem. First, the function is sufficiently well-behaved that any reasonable guess will lead to convergence for all three algorithms. Secondly, note that as x (time) approaches infinity, y approaches θ_1, so that an initial guess for θ_1 should be the maximum weight suspicioned for the animal. At time zero (birth), $y=\theta_1-\theta_2$, so θ_2 should be estimated by subtracting birth weight from the guess for θ_1. A little forethought leads to selecting θ_3 in the range from .01 to .1.

For the data in Table 3.1, the initial guess used was:

$$\theta_1^0=900, \qquad \theta_2^0=836, \qquad \theta_3^0=.05$$

The results for the three algorithms are summarized below:

	Mod. Gauss-Newton	Marquardt	Spiral
No. of iterations	9	5	6
SS Residual	307,763.8970870435	307,763.8969855355	307,763.8969043224
$\hat{\theta}_1$	800.11853	800.11771	800.12045
$\hat{\theta}_2$	768.57996	768.57663	768.57561
$\hat{\theta}_3$	.05594	.05594	.05594

There is not enough disparity among the three concerning the final residual sums of squares (SS Residual) or final parameter estimates, hence for this model the Marquardt procedure is preferred, having taken the fewest iterations. Note that $\hat{\theta}_1-\hat{\theta}_2$ does not estimate birth weight very well. The animal scientist may, then, reject these estimates and, thus, the model as unrealistic. It is possible to conduct the search under the constraint $\theta_1-\theta_2=$ birth weight. A procedure is discussed in Section 4.

Next we look at a textbook example which was selected so that computer results from the program NONLIN could be verified with published work. The model is ([1])

$$(3.2)\qquad y_i=\exp(-\theta_1 x_{1i}\exp(-\theta_2/x_{2i})\ )+\varepsilon_i$$

(a model of the fraction remaining at time x_1 of a chemical compound undergoing a first order reaction, where x_2 is the absolute temperature of the system. The equation is obtained as a solution to the differential equation

$$dy/dx = -ky.)$$

Data for the model is given in Table 3.2. Following [1], the initial guess is $\theta_1^0=750$, $\theta_2^0=1200$. Results for the three algorithms were:

	Mod. Gauss-Newton	Marquardt	Spiral
No. of iterations	9	10	11
SS Residual	.039806054412401	.039806054421774	.039806054415955
$\hat{\theta}_1$	813.87105	813.85866	813.86754
$\hat{\theta}_2$	961.00245	960.99876	961.00090

These results are in close agreement with [1]. It appears the objective function has a long narrow trough in the vicinity of $\hat{\underline{\theta}}$, so that Newtonian steps lead to quickest convergence, and any step toward either side of that direction is not as efficient.

The next example utilizes a more complicated function that is an intermediate result in work on segmenting two sigmoidal growth curves. More detail is found in Section 4, where this topic is discussed as an example of constrained estimation. The equation is

$$(3.3)\quad y_i=\begin{cases}\theta_1(1-\theta_3\exp(-\theta_2 x_i^2)\ )+\varepsilon_i\ , & x_i\leqq\gamma\\ \theta_1(1-\theta_3\exp(-\theta_2\gamma^2)+\frac{\theta_2\theta_3}{\theta_4}\exp(-\gamma^2\theta_2)(1-\exp(-\theta_4(x_i-\gamma^2)\ )+\varepsilon_i, & x_i>\gamma.\end{cases}$$

Rather than trying to fit this model to real data (which was done successfully after this example), we generate, with zero error, data using the values $\theta_1=.2$, $\theta_2=.004$, $\theta_3=.4$, $\theta_4=.009$, $\gamma=11.18$.

Table 3.2

Data for Example from Bard, Eq. 3.2

Experiment Number, i	Time x_{1_i} (hr)	Temperature x_{2_i} (°K)	Fraction remaining, y_i
1	0.1	100	0.980
2	0.2	100	0.983
3	0.3	100	0.955
4	0.4	100	0.979
5	0.5	100	0.993
6	0.05	200	0.626
7	0.1	200	0.544
8	0.15	200	0.455
9	0.2	200	0.225
10	0.25	200	0.167
11	0.02	300	0.566
12	0.04	300	0.317
13	0.06	300	0.034
14	0.08	300	0.016
15	0.1	300	0.066

The generated data is

x	y	x	y	x	y
1	.12032	8	.13807	15	.16428
2	.12127	9	.14214	16	.16641
3	.12283	10	.14367	17	.16811
4	.12496	11	.15069	18	.16945
5	.12761	12	.15487	19	.17046
6	.13073	13	.15853	20	.17123
7	.13424	14	.16166		

Using as an initial guess the values

$\theta_1^0=1.0$, $\theta_2^0=.01$, $\theta_3^0=1.0$, $\theta_4^0=.01$,

all three algorithms converged to $\underline{\hat{\theta}}=(.2,.004,.4,.009)'$ with zero residual sums of squares. The number of iterations necessary for convergence was 12, 13, and 7 for the modified Gauss-Newton, Marquardt, and Spiral algorithms, respectively, Thus the Spiral routine performed significantly better than the other two. An initial guess of

$\theta_1^0=5.0$, $\theta_2^0=.1$, $\theta_3^0=-2.0$, $\theta_4^0=.01$

did not lead to convergence for any of the three algorithms.

In our last example, from chemical engineering, we are trying to model reaction rate (y_i) as a function of temperature, ^{o}K, (x_{1_i}) nitrobenzene concentration (x_{2_i}), hydrogen concentration (x_{3_i}), and analine concentration (x_{4_i}). The equation is ([12])

$$(3.4) \qquad y_i=\frac{\theta_1\exp(-\theta_2/Rx_{1_i})x_{2_i}^{\theta_3}x_{3_i}^{\theta_4}}{1+\theta_5 x_{4_i}^{\theta_6}} .$$

Data for a particular experiment is given by Table 3.3. The thesis from which this example is obtained erroneously estimated the parameters, but we use the estimates provided there as "reasonable" guesses for starting values.

Table 3.3

Data for Chemical Engineering Example

(Eq. (3.4))

		Concentrations (gm-moles/cc)		
Reaction rate (y_i) (10^4gm-moles/min-gm)	Temp. x_{1_i} (°K)	Nitrobenzene $x_{2_i}(\times 10^8)$	Hydrogen $x_{3_i}(\times 10^7)$	Analine $x_{4_i}(\times 10^8)$
2.15	433	21.19	78.06	0
0.96	423	20.93	81.98	0
3.64	443	18.01	78.53	0
2.42	436	20.26	75.04	0
1.76	427.5	21.79	78.67	0
2.04	431	21.61	78.03	0
2.44	436	21.36	77.13	0
3.32	433	41.18	117.48	0
13.98	448	62.69	265.40	0
5.49	423	65.61	265.40	0
1.67	426	72.76	76.99	0
1.60	426	37.40	78.37	0
1.78	426	27.50	78.76	0
1.59	426	20.95	78.69	0
0.83	426	31.00	32.55	0
1.58	426	30.75	61.51	0
2.00	426	30.82	93.99	0
2.38	426	30.65	129.14	0
1.98	426	128.18	72.97	0
1.98	426	59.12	78.09	0
1.45	426	10.04	77.22	0
2.27	426	38.88	78.70	0
1.03	426	16.62	36.16	0
1.36	426	77.70	80.37	0
1.85	426	116.30	85.80	0
1.78	426	38.29	77.35	0
2.18	435	37.53	75.80	0
3.01	445	36.69	74.10	0
0.53	426	28.24	26.97	0
1.20	426	32.18	48.09	0
1.40	426	31.13	68.43	0
2.08	426	29.27	111.74	0
1.92	426	114.50	87.05	0
1.11	426	26.66	79.03	3.06
0.95	426	40.66	78.66	4.66
1.01	426	35.62	78.83	4.09
1.35	426	22.13	79.33	2.54
1.44	426	21.88	79.38	2.51
0.23	426	16.56	38.79	39.55
0.52	426	12.58	38.90	30.01
0.36	426	14.63	38.76	34.78
0.24	426	15.34	36.26	36.64

Using,

$\theta_1^0=190.50742$, $\theta_2^0=14719.64130$, $\theta_3^0=.05386$, $\theta_4^0=1.01018$, $\theta_5^0=.00421$, $\theta_6^0=.25189$, and R, the universal gas constant, 1.9869 cal/C° mole, we obtain the following results:

	Mod. Gauss-Newton	Marquardt	Spiral
No. of iterations	50*	50*	1†
SS Residual	209,844,885.6128651	212,585,078.6659825	218,569,377.1400358
$\hat{\theta}_1$	209.63174	191.3244	190.50742
$\hat{\theta}_2$	14,714.2779	14,714.51367	14,719.64130
$\hat{\theta}_3$	.04487	.05316	.05386
$\hat{\theta}_4$	1.01482	1.01043	1.01018
$\hat{\theta}_5$	.00001	.00030	.00471
$\hat{\theta}_6$	.56116	.37345	.23189

*No convergence after 50 iterations.
†No successful correction vector along path of steepest descent.

We cannot say that the Spiral algorithm led to convergence in view of the results of the other two procedures. Though neither the modified Gauss-Newton nor the Marquardt routines coverged in 50 iterations, we shall see below that the modified Gauss-Newton was near convergence.

To avoid endless loops, all three routines require some arbitrary stopping point in searching for a suitable correction vector. (This idea is not to be confused with stopping the routine because the new, updated estimate for $\underline{\theta}$ does not significantly differ from the previous estimate. Under this condition we say the procedure has converged.) When such a condition is reached, as was the case for the above starting value using the Spiral routine, NONLIN terminates the procedure with an appropriate message. When one encounters this condition, additional starting values should be tried to insure that a potential convergence point has indeed been reached. To illustrate, the following starting values were tried,

$\theta_1^0=195$, $\theta_2^0=14700$, $\theta_3^0=.0551$, $\theta_4^0=1.011$, $\theta_5^0=1.0$, $\theta_6^0=.05$

with results as follows:

	Mod. Gauss-Newton	Marquardt	Spiral
No. of iterations	35	50*	Did not converge.
SS Residual	209,844,915.1111224	215,846,225.5368481	
$\hat{\theta}_1$	209.57059	191.03260	
$\hat{\theta}_2$	14,714.27215	14,716.61624	
$\hat{\theta}_3$	.04489	.05345	
$\hat{\theta}_4$	1.01481	1.01031	
$\hat{\theta}_5$	.00001	.00202	
$\hat{\theta}_6$	.56173	.27874	

*No convergence after 50 iterations.

Using starting values:

$\theta_1^0=100$, $\theta_2^0=10{,}000$, $\theta_3^0=.01$, $\theta_4^0=1.0$, $\theta_5^0=.001$, $\theta_6^0=.5$,

the Marquardt and Spiral routines did not converge, but the modified Gauss-Newton converged to

$$\hat{\underline{\theta}}=(209.59451,\ 14{,}714.15046,\ .04487,\ 1.01482,\ .00001,\ .56111)'$$

in 34 iterations with a residual sums of squares equal to 209,844,879.8751693. Changing θ_3 and θ_4 to .05 caused all three routines to not converge. Additional starting values tried, none of which led to convergence, were:

$$(100,\ 1000,\ 0,\ 1,\ .05,\ .5)'$$

$$(10,\ 10000,\ .5,\ 2,\ 1,\ .01)'$$

$$(1000,\ 100,\ .25,\ 3,\ 5,\ 2)'$$

$$(1,\ 100,\ 1,\ 5,\ 10,\ 5)'.$$

It is apparent, now, that the starting value problem is critical. The problem is best handled by having prior information about the model and physical limitations of the parameters so that reasonable values can be obtained. Here the statistician, or modeler, must be in close contact with the experimenter, particularly for the more complicated nonlinear functions such as (3.4).

4. Constrained parameter estimation.

Outside factors, such as physical or management constraints, or the desire to test some hypothesis about $\underline{\theta}$ (e.g., see [3]), lead to the need for least squares estimates, $\hat{\underline{\theta}}$, in

$$\underline{y}=\underline{f}(\underline{x},\underline{\theta})+\underline{\varepsilon}$$

constrained by

$$g_1(\underline{\theta})=g_2(\underline{\theta})=\ldots=g_r(\underline{\theta})=0.$$

From the theory of Lagrangian multipliers, we can achieve least squares estimators for $\underline{\theta}$ by minimizing

$$(4.1)\qquad \Phi(\underline{\theta},\underline{\lambda})=\sum_{i=1}^{n}(y_i-f(\underline{x}_i,\underline{\theta}))^2+\sum_{s=1}^{r}2\lambda_s g_s(\underline{\theta})$$

with respect to $\underline{\theta}$ and $\underline{\lambda}$, an r×1 vector of multipliers. Differentiating (4.1) with respect to $\underline{\theta}$ and $\underline{\lambda}$ and setting the results equal to zero yields the p+r normal equations

$$(4.2)\qquad \frac{\partial\Phi(\underline{\theta},\underline{\lambda})}{\partial\theta_j}=-2\sum_{i=1}^{n}\{((y_i-f(\underline{x}_i,\underline{\theta}))\frac{\partial f(\underline{x}_i,\underline{\theta})\}}{\partial\theta_j}+2\sum_{s=1}^{r}\lambda_s\frac{\partial g_s(\underline{\theta})}{\partial\theta_j}=0;\quad j=1,\ldots,p$$

$$\frac{\partial\Phi(\underline{\theta},\underline{\lambda})}{\partial(2\lambda_s)}=g_s(\underline{\theta})=0;\quad s=i,\ldots,r.$$

Given some initial estimate of $\underline{\theta}$, say $\underline{\theta}^0$, expansion of $f(\underline{x}_i,\underline{\theta})$ in a Taylor series about $\underline{\theta}^0$ yields the first order approximation

$$f(\underline{x}_i,\underline{\theta})\doteq f(\underline{x}_i,\underline{\theta}^0)+\sum_{j=1}^{p}\frac{\partial f(\underline{x}_i,\underline{\theta}^0)d_j}{\partial\theta_j}\text{ , where}$$

$$d_j=(\theta_j-\tilde{\theta}_j)\text{ for some }\tilde{\theta}_j.$$

Similarly,
$$g_s(\underline{\theta})\doteq g_s(\underline{\theta}^0)+\sum_{j=1}^{p}\frac{\partial g_s(\underline{\theta}^0)d_j}{\partial\theta_j}.$$

Let $\underset{q\times 1}{\underline{d}} = [(d_r)]$; $\underset{r\times p}{\underline{G}} = \left[\frac{(\partial g_s(\underline{\theta}^0)\;)}{\partial \theta_j}\right]$; $\underline{g}^0=[(g_s(\underline{\theta}^0)\;)]$; and $\underline{\nu}$, $\underline{Z}$ be defined as before. Then upon substitution into (4.2), using our expanded approximations, we obtain

$$-2\underline{Z}'\underline{\nu}+2\underline{Z}'\underline{Z}\underline{d}+2\underline{G}'\underline{\lambda}=0$$

$$\underline{g}^0+\underline{G}\underline{d}=0 \;,$$

which can be rewritten as

(4.3)
$$\begin{bmatrix} \underline{Z}'\underline{Z} & \underline{G}' \\ \underline{G} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{d} \\ \underline{\lambda} \end{bmatrix} = \begin{bmatrix} \underline{Z}'\underline{\nu} \\ -\underline{g} \end{bmatrix} ,$$

a linear system from which a solution for $\underline{d}$ (our Gauss-Newton type correction vector) may be found. If we rewrite (4.3) as

$$\underline{W}\,\underline{\gamma} = \underline{h} \;,$$

with obvious notation, then $\underline{d}$ consists of the first p elements of $\underline{\gamma}$,where

$$\underline{\gamma} = \underline{W}^{-1}\,\underline{h} \;.$$

Once $\underline{d}$ is obtained, we compute $\underline{\theta}^1 = \underline{\theta}^0+\underline{d}$, as before, and we begin the process again, using $\underline{\theta}^1$ as our "initial guess".

If $g_s(\underline{\theta})$, for $s=1,\ldots,r$, is a linear function of $\underline{\theta}$, then $g_s(\underline{\theta}^j)=0$ will be satisfied at each iteration and thus we need only $\Phi(\underline{\theta}^{j+1})\leqq\Phi(\underline{\theta}^{\,j})$ as $\sum_{s=1}^{r} g_s(\underline{\theta}^j)=0$ in (4.1). If $g_s(\underline{\theta})$, for some s, is a nonlinear function of $\underline{\theta}$, then this is not the case, and we should verify that

$$\Phi(\underline{\theta}^{j+1},\underline{\lambda}^{j+1})\leqq\Phi(\underline{\theta}^j,\underline{\lambda}^j),$$

noting that (4.3) allows for solutions of $\underline{\lambda}$, as well as $\underline{d}$. The program NONLIN, described in the appendix, computes $\Phi(\underline{\theta}^{j+1})$ instead of $\Phi(\underline{\theta}^{j+1},\underline{\lambda}^{j+1})$, assuming $g_s(\underline{\theta}^{j+1})$ to be sufficiently close to zero. This has presented no problems thus far, as demonstrated by the next example, which involves fairly nonlinear constraints. The program allows the choice of a modified

Gauss-Newton or a Spiral type search for $\underline{\hat{\theta}}$ in the contsrained parameter space.

4.1 An example using growth curves.

Most growth curves are sigmoidal, i.e., symmetric, about some inflection point and thus are inappropriate to describe phenomena that are influenced by different factors at different times in their growth. Thus we may see a rapid growth in an animal until puberty, say, to be followed by a growth which is much less rapidly changing. A possible model for this type of behavior is to deseribe pre-puberty growth with one curve and maturity with another curve. Thus we are talking of the problem of segmenting two (or more) growth curves.

As an example, consider the dependent variable, y, to be a segmented function of two sigmoidal growth curves as

$$(4.4)\qquad y_i = \begin{cases} A_1(1-C_1\exp(-B_1x_i^2))+\varepsilon_i & \text{for } x_i \leqq \gamma \\ F+A_2(1-C_2\exp(-B_2x_i^2))+\varepsilon_i & \text{for } x_i > \gamma . \end{cases}$$

Realistically, we require continuity and differentiability of y at $x=\gamma$, giving the two constraints

$$(4.5)\qquad \begin{aligned} A_1(1-C_1\exp(-B_1\gamma^2)) &= F+A_2(1-C_2\exp(-B_2\gamma^2)) \\ 2A_1B_1C_1\gamma\exp(-B_1\gamma^2) &= 2A_2B_2C_2\gamma\exp(-B_2\gamma^2) . \end{aligned}$$

Equation (4.5) becomes

$$(4.6)\qquad \begin{aligned} F &= A_1(1-C_1\exp(-B_1\gamma^2))-A_2(1-C_2\exp(-B_2\gamma^2)) \\ A_2 &= \frac{A_1B_1C_1}{B_2C_2}\exp(-\gamma(B_1-B_2)). \end{aligned}$$

Substitution of (4.6) into (4.4) yields

$$(4.7) \qquad y_i = \begin{cases} A_1(1-C_1\exp(-B_1x_i^2)\ )+\varepsilon_i & x_i \leq \gamma, \\ A_1(1-C_1\exp(-B_1\gamma^2)+\dfrac{B_1C_1}{B_2}\ \exp(-\gamma^2B_1(1-\exp(-B_2(x_i^2-\gamma^2)\ )\ )\ +\varepsilon_i, & x_i > \gamma \end{cases}$$

which is (4.4) constrained by the continuity and differntiability restrictions of (4.5).

Suppose in (4.7) we let $x_i=i$, $i=1,\ldots,20$, $A_1=0.2$, $B_1=0.004$, $C_1=0.4$, and $B_2=0.009$. From [5] it is seen that for the growth curve

$$y_i=A_1(1-C_1\exp(-B_1x_i^2)\ ),$$

the inflection point is $1/\sqrt{2B_1}$. We wish to generate data from (4.7), but, for realism, we do not want the data to "flatten out" past the first inflection before beginning the part of the data produced by the second curve. Thus we fix $\gamma=1/\sqrt{2B_1}$ for generation. This is not necessary in the fitting of (4.7). If data actually contained a relative "plateau", the procedure should handle it; and in fact we will not restrict $\gamma=1/\sqrt{2B_1}$ in our search for γ, merely in the generation of the data. The data corresponding to (4.7) is that given after (3.3); in fact, (3.3) is (4.4) after reparameterizing to account for the constraints in (4.5). We should be able to estimate $\underline{\theta}$ with zero error, and this is the case using the program NONLIN (see discussion after (3.3)).

If our technique for constrained estimation is valid, we should be able to estimate all the parameters in (4.4) using the constraints of (4.5). Using the starting values

$A_1=.25$, $B_1=.01$, $C_1=.5$, $F=.1$, $A_2=.2$, $B_2=.01$, $C_2=.5$, $\gamma=15$,

the program NONLIN, using the modified Gauss-Newton option converged to

$\hat{A}_1=.2$, $\hat{B}_1=.004$, $\hat{C}_1=.4$, $\hat{F}=-.02706$, $\hat{A}_2=.20011$, $\hat{B}_2=.009$, $\hat{C}_2=.33196$, $\hat{\gamma}=11.18034$

in 18 iterations with zero residual error. Using the Spiral option, the routine failed to converge. Fixing γ=11.18034 and estimating the remaining seven parameters allowed both options to converge in 10 iterations to the values listed above with zero residual error.

References

1. Bard, Y. Nonlinear Parameter Estimation, Academic Press, New York,1974.

2. Box, G.E.P., and Kanemasu, H.."Topics in Model Building - Part II", Technical Report No. 321, Department of Statistics, University of Wisconsin, Madison, Wi.,1972.

3. DeBruin, R.L. "Testing Hypotheses for Nonlinear Models", Ph.D. Thesis, Kansas State University, Manhattan, Ks., 1971.

4. Draper, N.R., and Smith, H. Applied Regression Analysis, John Wiley and Sons, New York, 1966.

5. Grosenbaugh, L.R. "Generalization and Reparameterization of Some Sigmoid and Other Nonlinear Functions", Biometrics, Vol. ,708-714 (1965).

6. Hartley, H.O. "The Modified Gauss-Newton Method for the Fitting of Nonlinear Regression Functions by Least-squares", Technometrics, Vol. 3, 269-280 (1961).

7. Jones,A. "Spiral - A New Algorithm for Nonlinear Parameter Estimation Using Least Squares", The Computer Journal, Vol. 13, 301 (1970).

8. Marquardt, D.W. "An Algorithm for Least Squares Estimation of Nonlinear Parameters", Journal of the Society for Industrial and Applied Mathematics, Vol. 2, 431 (1963).

9. Milliken, G.A. A Theory of the Linear Model, unpublished class notes, Kansas State University, Manhattan, Ks., 1971.

10. Mood, A.M., Graybill, F.A., and Boes, D.C. Introduction to The Theory of Statistics, 3rd ed., McGraw-Hill Book Company, New York,1974.

11. Searle, S.R. Linear Models, John Wiley and Sons, New York, 1971.

12. Snyder, A.C. "Experimental Measurement and Simulation of Packed Bed Reactor Performance in the Presence of Catalyst Deactivation", Ph.D. Thesis, Kansas State University, Manhattan, Ks., 1974

APPENDIX

A User's Guide for NONLIN

The following control cards are needed to use the program NONLIN, which is documented following this guide:

1. Title card. Anything punched on the first control card (80 characters) is printed at the beginning of the output (once) for identification.

2. Limits card. Enter problem limitations as follows:

   cc 1-3: Number of observations (a value of the dependent variable along with its associated values of the independent variables constitute one observation).

   cc 5-6: Number of parameters to be estimated.

   cc 8-9: Number of independent variables in model.

   cc 11 : Number of method to be used,

   1-Modified Gauss-Newton

   2-Marquardt

   3-Spiral

   cc 13-14: Maximum number of iterations to be attempted.

   cc 16-17: Number of parameter constraints supplied.

3. Initial parameter estimates. Enter initial guesses sequentially as decimal numbers, using cc 1-10 for θ_1, cc 11-20 for θ_2, etc.; continue on next card if necessary. Use TRANS(B) subroutine to transform initial guesses if it is necessary to

rescale due to format limitations.

4. Format. FORTRAN format statement for data to follow, e.g., (F3.0,2X,3F4.0,1X,F10.0)

5. Data. Read in one observation at a time, with the dependent variable read in first.

SUBROUTINE TRANS(B): A user supplied FORTRAN subroutine (optional) in which the user can change (e.g., rescale) any of the parameters or variables read in. TRANS(B) is called once upon completion of reading in the data.

SUBROUTINE FUNC(B,F): A user supplied subroutine (mandatory) which gives the form of the model to be used. Parameters are in array B, independent variables are in array X. When more than one independent variable is present, all of the first variable is stored sequentially in X, followed by all of the second variable, etc. To address the i^{th} value of the third variable, address the $((2\times\text{no. of obs.})+i)^{th}$ member of X. The function is written as F( )=..., F containing the "predicted" value of Y using the current parameter estimates in B. An example of the model in Eq. (3.2) is given in SUBROUTINE FUNC(B,F) in the documentation following this guide.

SUBROUTINE LGRANG(B,G): A user supplied subroutine (optional) used to enter any parameter constraints. The constraints should be written as 0=(constraint). The first constraint is then entered as G(1)=(constraint), etc. As an example, to use the two constraints $\theta_1=\theta_2$ and $\theta_1=1/\theta_2$, enter G(1)=B(1)-B(2) and G(2)=B(1)*B(2)-1.

Output: The user supplied title, initial parameter estimates, limitations, and data, *as read in*, are printed on the first page of output. To print transformed estimates or data, include the appropriate WRITE statements in SUBROUTINE TRANS(B).

After each iteration, updated estimates of the parameters are printed along with the SS Residual using these estimates.

If the program terminates normally, the inverse of the sums of squares and cross products matrix $((\underline{Z}'\underline{Z})^{-1})$ is printed as evaluated at the final parameter estimates.

Limitations: The following limitations (maximums) are employed in NONLIN, as listed. They may be expanded by changing appropriate array sizes and formats. *If array sizes are altered, the user must be sure to change all affected arrays*.

No. of observations:	100
No. of parameters:	15
No. of indep. variables:	15
No. of constraints:	15

```
C   Y IS THE VECTOR OF OBSERVATIONS OF THE DEPENDENT VARIABLE
C   X IS THE MATRIX OF OBSERVATIONS OF THE INDEPENDENT VARIABLES
C   B IS THE VECTOR OF CURRENT PARAMETER ESTIMATES
C   F IS THE VECTOR OF FUNCTION VALUES USING CURRENT PARAMETER ESTIMATES
C   P IS THE MATRIX OF PARTIAL DERIVATIVES OF F EVALUATED AT B
C   A IS THE INVERSE OF P'P
C   PT IS THE TRANSPOSE OF P
C
      IMPLICIT REAL*8(A-H,O-$)
      REAL*8 LAMBDA
      COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
      DIMENSION TITLE(20),FRMT(20),F(100),B(15),P(1500),PT(1500),
     1A(225),L(15),M(15)
      LAMBDA=0.01D0
      READ (5,1000) TITLE
 1000 FORMAT (20A4)
      WRITE (6,1001) TITLE
 1001 FORMAT (1H1,T25,20A4)
      READ (5,1002) NOBS,NPARM,NVAR,NMETH,NITER,NCONST
 1002 FORMAT(I3,1X,I2,1X,I2,1X,I1,1X,I2,1X,I2)
C
C   NOBS- NUMBER OF OBSERVATIONS
C   NPARM- NUMBER OF PARAMETERS TO BE ESTIMATED
C   NVAR- NUMBER OF INDEPENDENT VARIABLES
C   NMETH- METHOD TO BE USED FOR NONLINEAR ESTIMATION
C    1- MODIFIED GAUSS-NEWTON (H.O. HARTLEY, TECHNOMETRICS, MAY 1961)
C    2- MARQUARDT (D.W. MARQUARDT, SIAM, 1963, P 461)
C    3- SPIRAL (A. JONES, COMPUTER JOURNAL, AUG, 1970)
C   NITER- MAXIMUM NUMBER OF ITERATIONS TO BE ATTEMPTED
C   NCONST- NUMBER OF PARAMETER CONSTRAINTS
C
      GO TO (101,102,103),NMETH
  101 WRITE(6,2000)
 2000 FORMAT('ONONLINEAR ESTIMATION USING MODIFIED GAUSS-NEWTON PROCEDUR
     1E.')
      GO TO 100
  102 WRITE(6,2001)
 2001 FORMAT('ONONLINEAR ESTIMATION USING MARQUARDT PROCEDURE.')
      GO TO 100
  103 WRITE(6,2002)
 2002 FORMAT('ONONLINEAR ESTIMATION USING SPIRAL PROCEDURE.')
  100 WRITE(6,2003)NOBS,NPARM,NVAR,NITER,NCONST
 2003 FORMAT('ONO. OF OBS.= ',I3,5X,'NO. OF PARAMETERS= ',I2,5X,
     1'NO. OF INDEP. VAR.= ',I2/' MAXIMUM NO. OF ITERATIONS= ',I2,2X,
     2 'NO. OF USER-SUPPLIED CONSTRAINTS= ',I2)
      READ (5,1003)(B(I),I=1,NPARM)
 1003 FORMAT(8F10.5)
      WRITE(6,2004)
 2004 FORMAT('OORIGINAL PARAMETER ESTIMATES.')
      WRITE(6,2005)(I,I=1,NPARM)
 2005 FORMAT(12X,I2,5(18X,I2))
      WRITE(6,2006)(B(I),I=1,NPARM)
 2006 FORMAT(1X,6F20.5)
      READ (5,1004) FRMT
 1004 FORMAT(20A4)
      WRITE (6,1005)
 1005 FORMAT(32HODATA AS INPUT TO PROGRAM NONLIN)
      DO 2 I=1,NOBS
```

```
                N=(NVAR-1)*NOBS+I
      C
      C   DEPENDENT VARIABLE IS READ IN FIRST
      C   IF MORE THAN CNE INDEPENDENT VARIABLE, MATRIX X IS STORED AS A STRING
      C
                READ(5,FRMT)Y(I),(X(J),J=I,N,NOBS)
              2 WRITE(6,2006)Y(I),(X(J),J=I,N,NOBS)
      C
      C   USE TRANS(B) TO FORM ANY NEW VARIABLES, OR CHANGE VALUES OF VARIABLES WHICH
      C   ARE BEYOND FORMATS. TRANS(B) IS CALLED CNLY ONCE.
      C
                CALL TRANS(B)
                CALL FUNC(B,F)
                SS=SSQS(F)
                WRITE(6,1006) SS
           1006 FORMAT(1H0,23HINITIAL SUM OF SQUARES=,1X,F40.15)
                N=NPARM*NPARM
                ITER=0
              5 ITER=ITER+1
                IF(ITER.GT.NITER) GO TO 70
                CALL DERIV(P,F,B)
                IERR3=1
                ITERM=0
                GO TO (10,20,30),NMETH
             10 CALL GAUSS(P,F,B,IERR3)
                GO TO 40
             20 CALL MQDT(P,B,LAMBDA,F)
                GO TO 40
             30 CALL SPIRAL(P,F,B,IERR3)
      C
      C   CHECK FOR CONVERGENCE
      C
             40 DO 50 I=1,NPARM
                IF((DABS(DELTA(I))/(.001D0+DABS(B(I)))).GT.1.D-6) GO TO 51
             50 CONTINUE
                GO TO 52
             51 ITERM=1
             52 WRITE(6,1007) ITER
           1007 FORMAT(1H0,T20,13HITERATICN NO.,I3)
                WRITE(6,1008)
           1008 FORMAT(1X,16HPARAMETER VALUES)
                WRITE(6,1009)(I,I=1,NPARM)
           1009 FORMAT(12X,I2,5(18X,I2))
                WRITE(6,1010)(B(I),I=1,NPARM)
           1010 FORMAT(1X,6(F20.5))
                WRITE(6,1011) SS
           1011 FORMAT(' SUM OF SQUARES= ',F40.15)
      C
      C   IERR3 POINTS TO APPROPRIATE TERMINATION MESSAGE
      C
                GO TO (55,75,71),IERR3
             55 IF(ITERM.GT.0) GO TO 5
             60 WRITE(6,1012)
           1012 FORMAT(92H0ESTIMATION PROCEDURE TERMINATED DUE TO APPARENT CONVERG
               1ENCE TO THE LISTED PARAMETER VALUES.)
                GO TO 80
             70 WRITE(6,1013) NITER
           1013 FORMAT(33H0PROCEDURE DOESN'T CCNVERGE AFTER,I3,12H ITERATIONS.)
      GO TO 80
   71 WRITE(6,1015)
 1015 FORMAT(' PROCEDURE TERMINATED. NO SUCCESSFUL CORRECTION VECTOR FOU
     1ND ALONG PATH OF STEEPEST DESCENT.'/' POSSIBLE CONVERGENCE POINT H
     2AS BEEN REACHED.')
      GO TO 80
   75 WRITE(6,1014)
 1014 FORMAT('OPROGRAM TERMINATED. NO SUCCESSFUL CORRECTION VECTOR FOUND
     1 ALONG TAYLOR-SERIES PATH. POSSIBLE CONVERGENCE POINT HAS BEEN REA
     2CHED.')
C
C  COMPUTE Z'Z INVERSE AND PRINT
C
   80 CALL FUNC(B,F)
      CALL DERIV(P,F,B)
      JI=0
      DO 90 I=1,NOBS
      IJ=I-NOBS
      DO 90 J=1,NPARM
      IJ=IJ+NOBS
      JI=JI+1
   90 PT(JI)=P(IJ)
      CALL MMULT(A,PT,P,NPARM,NOBS,NPARM)
      CALL DMINV(A,NPARM,D,L,M)
      WRITE(6,1016)
 1016 FORMAT('OINVERSE OF SS & CROSS PRODUCTS MATRIX OF PARTIALS EVALUAT
     1ED AT FINAL B:'/)
      K=(NPARM-1)*NPARM
      DO 91 I=1,NPARM
      N=K+I
   91 WRITE(6,1017)(A(J),J=I,N,NPARM)
 1017 FORMAT(1X,10E12.5)
      STOP
      END
```

```
FORTRAN IV G LEVEL  21                SSQS               DATE = 76005          21/13/08

              FUNCTION SSCS(F)
              IMPLICIT REAL*8(A-H,O-$)
              COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
              DIMENSICN F(NOBS)
C
C     SUBRCUTINE CCMPUTES ERROR SUM OF SQUARES FOR A PARTICULAR VALUE OF B
C
              CALL ERRSET(207,256,-1,1)
              SSQS=0.0
              DO 1 I=1,NCBS
              SSQS=SSQS+(Y(I)-F(I))*(Y(I)-F(I))
              CALL OVERFL(J)
              IF(J.EQ.1) GO TO 2
            1 CONTINUE
              GO TO 3
            2 SSQS=1.D74
            3 CALL ERRSET(207,256,256,2)
              RETURN
              END
```

```
FORTRAN IV G LEVEL  21                DERIV            DATE = 76005         21/13/08

      SUBROUTINE DERIV(P,F,B)
      IMPLICIT RFAL*8(A-H,O-$)
      COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
      DIMENSICN B(15),BO(15),F(100),FO(100),P(1500)
C
C   SUBROUTINE CCMPUTES MATRIX OF PARTIAL DERIVATIVES, P
C
      DO 3  I=1,NPARM
      DO 1  J=1,NPARM
    1 BO(J)=B(J)
      H=B(I)*.001DO
      H1=1.DO/H
      BO(I)=B(I)+H
      CALL  FUNC(BO,FO)
      DO 2  J=1,NOBS
    2 P((I-1)*NOBS+J)=(FO(J)-F(J))*H1
    3 CCNTINUE
      RETURN
      END
```

```
FORTRAN IV G LEVEL  21                  MMULT                  DATE = 76005          21/13/08

                  SUBROUTINE MMULT(AB,A,B,L,M,N)
                  IMPLICIT REAL*8(A-H,O-$)
                  DIMENSION AB(L,N),A(L,M),B(M,N)
                  DO 3 I=1,L
                  DO 2 J=1,N
                  AB(I,J)=0.0
                  DO 1 K=1,M
                1 AB(I,J)=A(I,K)*B(K,J)+AB(I,J)
                2 CONTINUE
                3 CONTINUE
                  RETURN
                  END
```

```
      SUBROUTINE GAUSS(P,F,B,IERR3)
      IMPLICIT REAL*8(A-H,O-$)
      COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
      DIMENSION P(1500),F(100),B(15),A(225),PT(1500),YF(100),TEMP(30),
     1 L(30),M(30),B1(15),B2(15),W(900),G(15),GDERIV(225),DELTA1(30)
C
C    FOLLOWING ROUTINE CALCULATES PT FROM P
C
      JI=0
      DO 1 I=1,NOBS
      IJ=I-NOBS
      DO 1 J=1,NPARM
      IJ=IJ+NOBS
      JI=JI+1
    1 PT(JI)=P(IJ)
C
C    YF IS Y-F
C
      DO 5 I=1,NCBS
    5 YF(I)=Y(I)-F(I)
      CALL MMULT(TEMP,PT,YF,NPARM,NOBS,1)
C
C    A=PT*P
C
      CALL MMULT(A,PT,P,NPARM,NOBS,NPARM)
      IF(NCONST.EQ.0) GO TO 180
C
C    IF NCONST NOT 0, NEED TO BUILD AUGMENTED MATRIX
C
      N=NPARM+NCCNST
      N2=N*N
      DO 100 I=1,N2
  100 W(I)=0.D0
      CALL LGRANG(B,G)
      CALL LGRDRV(B,G,GDERIV)
      DO 130 I=1,NPARM
      K=(I-1)*N
      K1=(I-1)*NPARM
      K2=(I-1)*NCCNST
      DO 110 J=1,NPARM
  110 W(K+J)=A(K1+J)
      K=K+NPARM
      DO 120 J=1,NCONST
  120 W(K+J)=GDERIV(K2+J)
  130 CONTINUE
      K=NPARM*N
      DO 150 I=1,NCONST
      K1=(I-1)*NPARM
      K2=(I-1)*N
      DO 140 J=1,NPARM
  140 W(K+K2+J)=GDERIV((J-1)*NCONST+I)
  150 CONTINUE
      WRITE(6,3000)
      DO 151 I=1,N
      N1=(N-1)*N+I
C
C    WRITE AUGMENTED MATRIX, VALUE CF CONSTRAINTS, AND LAGRANGE MULTIPLIERS SO
C    USER CAN CHFCK IF CONSTRAINTS ARE NEAR ZERO AT CCNVERGENCE
```

```
C
  151 WRITE(6,3001)(W(J),J=I,N1,N)
 3000 FORMAT(//' AUGMENTED MATRIX = '/)
 3001 FORMAT(1X,9(1X,E12.5))
      CALL DMINV(W,N,D,L,M)
      DO 160 I=1,NCONST
      WRITE(6,4000)I,G(I)
 4000 FORMAT('0VALUE OF CONSTRAINT NO. ',I2,' = ',E12.5)
  160 TEMP(NPARM+I)=-G(I)
      CALL MMULT(DELTA1,W,TEMP,N,N,1)
      DO 170 I=1,NPARM
  170 DELTA(I)=DELTA1(I)
      WRITE(6,4001)(DELTA1(K+NPARM), K=1,NCONST)
 4001 FORMAT('0LAMBDA= '/1X,9E12.5)
      GO TO 190
  180 CALL DMINV(A,NPARM,D,L,M)
C
C    DELTA=(PT*P)INVERSE*PT*(Y-F) (SEE DRAPER & SMITH, 1966, P 268)
C
      CALL MMULT(DELTA,A,TEMP,NPARM,NPARM,1)
  190 V=1.D0
    6 V1=.5D0*V
      DO 10 I=1,NPARM
      B1(I)=B(I)+V1*DELTA(I)
   10 B2(I)=B(I)+V*DELTA(I)
      CALL FUNC(B1,F)
      SS1=SSQS(F)
      CALL FUNC(B2,F)
      SS2=SSQS(F)
      V1=(.5D0+.25D0*(SS-SS2)/(SS2-SS1-SS1+SS))*V
      DO 20 I=1,NPARM
   20 B1(I)=B(I)+V1*DELTA(I)
      CALL FUNC(B1,F)
      SS1=SSQS(F)
      IF(SS1.LT.SS) GO TO 30
      IF(V.LT..01D0) GO TO 50
      V=.5D0*V
      GO TO 6
   30 DO 40 I=1,NPARM
   40 B(I)=B1(I)
      SS=SS1
      GO TO 60
   50 IERR3=2
   60 RETURN
      END
```

```
FORTRAN IV G LEVEL  21                    MQDT                  DATE = 76005          21/13/08

                  SUBROUTINE MQDT(P,B,LAMBDA,F)
                  IMPLICIT REAL*8(A-H,O-$)
                  REAL*8 LAMBCA,LAM1
                  CCMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
                  DIMENSION P(1500),F(100),B(15),A(225),PT(1500),TEMP(15),YF(100),
                 1AOLD(225),B1(15),B2(15)
         C
         C    FORM SSCP MATRIX OF PARTIALS AND STORE IN A
         C
                  JI=0
                  DO 1 I=1,NCBS
                  IJ=I-NOBS
                  DO 1 J=1,NPARM
                  IJ=IJ+NOBS
                  JI=JI+1
                1 PT(JI)=P(IJ)
                  CALL MMULT(A,PT,P,NPARM,NCBS,NPARM)
                  DO 2 I=1,NOBS
                2 YF(I)=Y(I)-F(I)
                  CALL MMULT(TEMP,PT,YF,NPARM,NOBS,1)
                  DO 3 I=1,N
                3 AOLD(I)=A(I)
         C
         C    PERFORM SCALE TRANSFORMATION ON MATRIX OF PARTIALS, A
         C
                  DO 4 I=1,NPARM
                  II=(I-1)*NFARM
                  RCOT=DSQRT(ACLD(II+I))
                  TEMP(I)=TEMP(I)/ROOT
                  DO 4 K=1,NPARM
                  IJ=II+K
                4 A(IJ)=AOLD(IJ)/(ROOT*DSQRT(AOLD((K-1)*NPARM+K)))
         C
         C    BEGIN MARQUARDT ITERATIVE PROCEDURE
         C
                  LAM1=LAMBDA*0.1D0
                  CALL MQDT2(A,B,B2,TEMP,LAM1,SS2,F,AOLD)
                  IF(SS2.LE.SS) GO TO 10
                  CALL MQDT2(A,B,B1,TEMP,LAMBCA,SS1,F,ACLD)
                  IF(SS1.LE.SS) GO TO 7
                5 LAMBDA=LAMBCA*10.D0
                  CALL MQDT2(A,B,B1,TEMP,LAMBDA,SS1,F,AOLD)
                  IF(SS1.LE.SS) GO TO 7
                  GO TO 5
               10 SS=SS2
                  LAMBCA=LAM1
                  DO 6 I=1,NPARM
                6 B(I)=B2(I)
                  GO TO 9
                7 SS=SS1
                  DO 8 I=1,NPARM
                8 B(I)=B1(I)
                9 RETURN
                  END
```

```
FORTRAN IV G LEVEL  21                  MQDT2               DATE = 76005          21/13/08

      SUBROUTINE MQDT2(A,B,B1,TEMP,LAMBDA,SSQ,F,AOLD)
      IMPLICIT REAL*8(A-H,O-$)
      REAL*8 LAMBDA,LAM1
      COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
      DIMENSION A(225),F(100),B(15),B1(15),TEMP(15),AI(225),L(15),M(15),
     1 AOLD(225)
      DO 10 I=1,N
   10 AI(I)=A(I)
      DO 1 I=1,NPARM
      II=(I-1)*NPARM+I
    1 AI(II)=AI(II)+LAMBDA
      CALL DMINV(AI,NPARM,D,L,M)
      CALL MMULT(DELTA,AI,TEMP,NPARM,NPARM,1)
C
C    RESCALE CORRECTION VECTOR, DELTA
C
      DO 4 I=1,NPARM
    4 DELTA(I)=DELTA(I)/DSQRT(AOLD((I-1)*NPARM+I))
      DO 5 I=1,NPARM
    5 B1(I)=B(I)+DELTA(I)
      CALL FUNC(B1,F)
      SSQ=SSQS(F)
      RETURN
      END
```

```
FORTRAN IV G LEVEL  21                  SPIRAL               DATE = 76005             21/13/08

       SUBROUTINE SPIRAL(P,F,B,IERR3)
       IMPLICIT REAL*8(A-H,O-$)
       COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
       DIMENSION P(1500),PT(1500),A(225),F(100),YF(100),B(15),TEMP(30),
      1B1(15),S(15),SSQ(10),W(900),G(15),GDERIV(225),DELTA1(30),L(30),
      2M(30)
       REAL*8 U(10)/.1D0,.1818181818181818,.307692307922306,
      1.470588235294115,.64D0,.780487804878049,.876712328767122,
      2.934306569343064,.966037735849055,.982725527831093/
C
C    INITIAL PART CF ROUTINE IS SIMILAR TC GAUSS ROUTINE
C
       JI=0
       DO 1 I=1,NOBS
       IJ=I-NOBS
       DO 1 J=1,NPARM
       IJ=IJ+NOBS
       JI=JI+1
     1 PT(JI)=P(IJ)
       CALL MMULT(A,PT,P,NPARM,NOBS,NPARM)
       DO 2 I=1,NCBS
     2 YF(I)=Y(I)-F(I)
       CALL MMULT(TEMP,PT,YF,NPARM,NOBS,1)
       IF (NCCNST.EQ.0) GO TO 3
       N=NCONST+NPARM
       N2=N*N
       DO 1100 I=1,N2
  1100 W(I)=0.D0
       CALL LGRANG(B,G)
       CALL LGRDRV(B,G,GDERIV)
       DO 1130 I=1,NPARM
       K=(I-1)*N
       K2=(I-1)*NCCNST
       K1=(I-1)*NPARM
       DO 1110 J=1,NPARM
  1110 W(K+J)=A(K1+J)
       K=K+NPARM
       DO 1120 J=1,NCONST
  1120 W(K+J)=GDERIV(K2+J)
  1130 CONTINUE
       K=NPARM*N
       DO 1150 I=1,NCCNST
       K1=(I-1)*NPARM
       K2=(I-1)*N
       DO 1140 J=1,NPARM
  1140 W(K+K2+J)=GDERIV((J-1)*NCCNST+I)
  1150 CONTINUE
       WRITE(6,3000)
       DO 1151 I=1,N
       N1=(N-1)*N+I
  1151 WRITE(6,3001)(W(J),J=I,N1,N)
  3000 FORMAT(//' AUGMENTED MATRIX = '/)
  3001 FCRMAT(1X,9(1X,E12.5))
       CALL DMINV(W,N,D,L,M)
       DO 1160 I=1,NCONST
       WRITE(6,4000) I,G(I)
  4000 FORMAT('OVALUE OF CCNSTRAINT NO. ',I2,' = ',E12.5)
  1160 TEMP(NPARM+I)=-G(I)
```

```
      CALL MMULT(DELTA1,W,TEMP,N,N,1)
      DO 1170 I=1,NPARM
 1170 DELTA(I)=DELTA1(I)
      WRITE(6,4001)(DELTA1(K+NPARM),K=1,NCONST)
 4001 FORMAT('CLAMBDA= '/1X,9E12.5)
      GO TO 4
    3 CALL MMULT(DELTA,A ,TEMP,NPARM,NPARM,1)
      CALL DMINV(A,NPARM,D,L,M)
    4 IFLAG=0
    5 DO 10 I=1,NPARM
   10 B1(I)=B(I)+DELTA(I)
      CALL FUNC(B1,F)
      SS2=SSQS(F)
      IF(SS2.LT.SS) GO TO 40
      DO 20 I=1,NPARM
   20 B1(I)=B(I)+.5D0*DELTA(I)
      CALL FUNC(B1,F)
      SS1=SSQS(F)
      IF(SS1.LT.SS) GO TO 50
      V=.5D0+.25D0*(SS-SS2)/(SS2+SS-SS1-SS1)
      DO 30 I=1,NPARM
   30 B1(I)=B(I)+V*DELTA(I)
      CALL FUNC(B1,F)
      SS1=SSQS(F)
      IF(SS1.LT.SS) GO TO 50
      GO TO 80
   40 SS=SS2
      GO TO 60
   50 SS=SS1
   60 DO 70 I=1,NPARM
      DELTA(I)=B1(I)-B(I)
   70 B(I)=B1(I)
      GO TO 230
C
C    BEGIN SEARCH ALONG THE SPIRAL
C
   80 T=0.D0
      GRAD=0.D0
      DO 85 I=1,NPARM
   85 DELTA(I)=DELTA(I)*V
C
C    TEMP VECTOR LIES IN DIRECTION OF STEEPEST DESCENT. NEED TO GET LENGTH OF
C    TEMP & RESCALE TO LENGTH OF DELTA WHICH IS GAUSS-NEWTON CORRECTION
C
   89 DO 90 I=1,NPARM
      T=T+DELTA(I)*DELTA(I)
   90 GRAD=GRAD+TEMP(I)*TEMP(I)
      TD=DSQRT(T)/DSQRT(GRAD)
      DO 100 I=1,NPARM
  100 TEMP(I)=-TEMP(I)*TD
      TMIND=0.D0
C
C    THIS SECTION IS STRAIGHTFORWARD APPLICATION OF SPIRAL ALGORITHM AS
C    PUBLISHED BY JONES
C
      DO 110 I=1,NPARM
  110 TMIND=TMIND+(TEMP(I)-DELTA(I))*(TEMP(I)-DELTA(I))
      COSG=1.D0-((TMIND)/(T+T))
      GAMMA=DARCOS(COSG)
      SING=DSIN(GAMMA)
      I=0
  120 I=I+1
      CALL SPRL2(TEMP,B,B1,U(I),V1,SSQ(I),GAMMA,SING,COSG,S,F)
      SS1=SSQ(I)
      IF(SS1.LT.SS) GO TO 200
      IF(I.LT.3) GO TO 120
      I1=I-1
      I2=I-2
      IF(SSQ(I1).GT.SSQ(I2).OR.SSQ(I1).GT.SS1) GO TO 130
      U1=U(I2)
      U2=U(I1)
      U3=U(I)
      U4=U1-U2
      U5=U1*U1-U2*U2
      U6=U1-U3
      U7=U1*U1-U3*U3
      U8=((SSQ(I2)-SSQ(I1))*U7-(SSQ(I2)-SS1)*U5)/(U4*U7-U6*U5)
      U8=-U8*U5/(SSQ(I2)-SSQ(I1)-U6*U4)
      CALL SPRL2(TEMP,B,B1,U8,V1,SS1,GAMMA,SING,COSG,S,F)
      IF(SS1.LT.SS) GO TO 200
  130 CONTINUE
      IF(U(I+1).LT..97D0) GO TO 120
      IFLAG=IFLAG+1
      IF(IFLAG.GT.3) GO TO 145
C
C   IF NO SUCCESSFUL POINT IS FOUND, HALVE THE LENGTH OF DELTA & REPEAT
C   (UP TO 3 TIMES)
C
      DO 140 I=1,NPARM
  140 DELTA(I)=.5D0*DELTA(I)
      GO TO 89
  145 SLONG=0
      DO 150 I=1,NPARM
  150 SLONG=SLONG+S(I)*S(I)
      TD=1.D0/TD
      SG=TD*DSQRT(SLONG)/DSQRT(GRAD)
      DO 160 I=1,NPARM
  160 TEMP(I)=TEMP(I)*SG
  170 DO 180 I=1,NPARM
      TEMP(I)=.5D0*TEMP(I)
  180 B1(I)=B(I)+TEMP(I)
      CALL FUNC(B1,F)
      SS1=SSQS(F)
      IF(SS1.LT.SS) GO TO 200
      DO 190 I=1,NPARM
      IF(TEMP(I).GT.1.D-6) GO TO 170
  190 CONTINUE
      GO TO 220
  200 DO 210 I=1,NPARM
      DELTA(I)=B1(I)-B(I)
  210 B(I)=B1(I)
      SS=SS1
      GO TO 230
  220 IERR3=3
  230 RETURN
      END
```

```
FORTRAN IV G LEVEL  21                    SPRL2               DATE = 76005          21/13/08

      SUBROUTINE SPRL2(TEMP,B,B1,U,V1,SS1,GAMMA,SING,COSG,S,F)
      IMPLICIT REAL*8(A-H,O-$)
      COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
      DIMENSION TEMP(15),B(15),B1(15),S(15),F(100)
C
C    THIS SUBROUTINE COMPUTES COORDINATES OF POINT ON SPIRAL TO BE SEARCHED.
C    SEE JONES FOR DETAIL
C
      BINCR=.5D0
      DO 1 I=1,NPARM
      B1(I)=B(I)+TEMF(I)
    1 S(I)=B(I)+DELTA(I)
      THETA=DATAN((U*SING)/(1.D0-U+U*COSG))
    3 COSB=DCOS(GAMMA*BINCR)
      TG=THETA/GAMMA
      T1=(1.D0-THETA*COSB-(1.D0-GAMMA*COSB)*TG*TG)
      XSI=U*SING/DSIN(THETA)
      XSI=T1/XSI
      DO 120 I=1,NPARM
      S(I)=XSI*(L*TEMP(I)+(1.D0-U)*DELTA(I))
  120 B1(I)=B(I)+S(I)
      CALL FUNC(B1,F)
      SS1=SSQS(F)
      IF(SS1.LT.1.D74) GO TO 2
C
C    GET HERE WHEN WE HAVE OVERFLOWED SSQS FUNCTION. NEED TO DECREASE
C    INCIDENT ANGLE OF SPIRAL
C
      BINCR=BINCR*.5D0
      GO TO 3
    2 RETURN
      END
```

```
FORTRAN IV G LEVEL  21                  MAIN                DATE = 76005          21/13/08
              C                                                                    MINV 003
              C       SUBROUTINE MINV   DMINV                                      MINV 004
              C                                                                    MINV 005
              C       PURPCSE                                                      MINV 006
              C          INVERT A MATRIX                                           MINV 007
              C                                                                    MINV 008
              C       USAGE                                                        MINV 009
              C          CALL MINV(A,N,D,L,M)                                      MINV 010
              C                                                                    MINV 011
              C       DESCRIPTICN OF PARAMETERS                                    MINV 012
              C          A - INPUT MATRIX, DESTRCYED IN COMPUTATION AND REPLACED BY  MINV 013
              C              RESULTANT INVERSE.                                    MINV 014
              C          D - RESULTANT DETERMINANT                                 MINV 01
              C          L - WORK VECTOR OF LENGTH N                               MINV 017
              C          M - WCRK VECTOR OF LENGTH N                               MINV 018
              C                                                                    MINV 019
              C       REMARKS                                                      MINV 020
              C          MATRIX A MUST BE A GENERAL MATRIX                         MINV 021
              C                                                                    MINV 022
              C       SUBROUTINES AND FUNCTICN SUBPROGRAMS REQUIRED                MINV 023
              C          NONE                                                      MINV 024
              C                                                                    MINV 025
              C       METHOD                                                       MINV 026
              C          THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT   MINV 027
              C          IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT  MINV 028
              C          THE MATRIX IS SINGULAR.                                   MINV 029
              C                                                                    MINV 030
              C     ...............................................................MINV 031
              C                                                                    MINV 032
0001                SUBROUTINE DMINV(A,N,D,L,M)
0002                DIMENSION A(225),L(15),M(15)
              C                                                                    MINV 035
              C       .............................................................MINV 036
              C                                                                    MINV 037
              C       IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE  MINV 038
              C       C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION    MINV 039
              C       STATEMENT WHICH FOLLCWS.                                     MINV 040
              C                                                                    MINV 041
0003                DOUBLE PRECISICN A,D,BIGA,HOLD,DABS                            MINV 042
              C                                                                    MINV 043
              C       APPEARING IN OTHER ROUTINES USED IN CONJUNCTICN WITH THIS    MINV 045
              C       ROUTINE.                                                     MINV 046
              C                                                                    MINV 047
              C       THE DOUBLE PRECISION VERSICN OF THIS SUBROUTINE MUST ALSO    MINV 048
              C       CONTAIN DOUBLE PRECISICN FORTRAN FUNCTIONS.  ABS IN STATEMENT  MINV 049
              C       10 MUST BE CHANGED TO DABS.                                  MINV 050
              C                                                                    MINV 051
              C       .............................................................MINV 052
              C                                                                    MINV 053
              C       SEARCH FCR LARGEST ELEMENT                                   MINV 054
              C                                                                    MINV 055
0004                D=1.0                                                          MINV 056
0005                NK=-N                                                          MINV 057
0006                DO 80 K=1,N                                                    MINV 058
0007                NK=NK+N                                                        MINV 059
0008                L(K)=K                                                         MINV 060
0009                M(K)=K                                                         MINV 061
0010                KK=NK+K                                                        MINV 062
```

```
0011                BIGA=A(KK)                                                        MINV 063
0012                DO 20 J=K,N                                                       MINV 064
0013                IZ=N*(J-1)                                                        MINV 065
0014                DO 20 I=K,N                                                       MINV 066
0015                IJ=IZ+I                                                           MINV 067
0016             10 IF(DABS(BIGA)-DABS(A(IJ))) 15,20,20                               MINV 068
0017             15 BIGA=A(IJ)                                                        MINV 069
0018                L(K)=I                                                            MINV 070
0019                M(K)=J                                                            MINV 071
0020             20 CONTINUE                                                          MINV 072
           C                                                                          MINV 073
           C           INTERCHANGE RCWS                                               MINV 074
           C                                                                          MINV 075
0021                J=L(K)                                                            MINV 076
0022                IF(J-K) 35,35,25                                                  MINV 077
0023             25 KI=K-N                                                            MINV 078
0024                DO 30 I=1,N                                                       MINV 079
0025                KI=KI+N                                                           MINV 080
0026                HOLD=-A(KI)                                                       MINV 081
0027                JI=KI-K+J                                                         MINV 082
0028                A(KI)=A(JI)                                                       MINV 083
0029             30 A(JI) =HOLD                                                       MINV 084
           C                                                                          MINV 085
           C           INTERCHANGE COLUMNS                                            MINV 086
           C                                                                          MINV 097
0030             35 I=M(K)                                                            MINV 088
0031                IF(I-K) 45,45,38                                                  MINV 089
0032             38 JP=N*(I-1)                                                        MINV 090
0033                DO 40 J=1,N                                                       MINV 091
0034                JK=NK+J                                                           MINV 092
0035                JI=JP+J                                                           MINV 093
0036                HOLD=-A(JK)                                                       MINV 094
0037                A(JK)=A(JI)                                                       MINV 095
0038             40 A(JI) =HOLD                                                       MINV 096
           C                                                                          MINV 097
           C           DIVIDE COLUMN BY MINUS PIVCT (VALUE OF PIVOT ELEMENT IS        MINV 098
           C           CONTAINED IN BIGA)                                             MINV 099
           C                                                                          MINV 100
0039             45 IF(BIGA) 48,46,48                                                 MINV 101
0040             46 D=0.0                                                             MINV 102
0041                RETURN                                                            MINV 103
0042             48 DO 55 I=1,N                                                       MINV 104
0043                IF(I-K) 50,55,50                                                  MINV 105
0044             50 IK=NK+I                                                           MINV 106
0045                A(IK)=A(IK)/(-BIGA)                                               MINV 107
0046             55 CONTINUE                                                          MINV 108
           C                                                                          MINV 109
           C           REDUCE MATRIX                                                  MINV 110
           C                                                                          MINV 111
0047                DO 65 I=1,N                                                       MINV 112
0048                IK=NK+I                                                           MINV 113
0049                HOLD=A(IK)                                                        MINV M01
0050                IJ=I-N                                                            MINV 114
0051                DO 65 J=1,N                                                       MINV 115
0052                IJ=IJ+N                                                           MINV 116
0053                IF(I-K) 60,65,60                                                  MINV 117
0054             60 IF(J-K) 62,65,62                                                  MINV 118
0055             62 KJ=IJ-I+K                                                         MINV 119
```

```
FORTRAN IV G LEVEL  21                DMINV              DATE = 76005          21/13/08

 0056                A(IJ)=HOLD*A(KJ)+A(IJ)                                          MINV M02
 0057             65 CONTINUE                                                        MINV 121
           C                                                                         MINV 122
           C            DIVIDE ROW BY PIVOT                                          MINV 123
           C                                                                         MINV 124
 0058                KJ=K-N                                                          MINV 125
 0059                DO 75 J=1,N                                                     MINV 126
 0060                KJ=KJ+N                                                         MINV 127
 0061                IF(J-K) 70,75,70                                                MINV 128
 0062             70 A(KJ)=A(KJ)/BIGA                                                MINV 129
 0063             75 CONTINUE                                                        MINV 130
           C                                                                         MINV 131
           C            PRODUCT OF PIVOTS                                            MINV 132
           C                                                                         MINV 133
 0064                D=D*BIGA                                                        MINV 134
           C                                                                         MINV 135
           C            REPLACE PIVOT BY RECIPROCAL                                  MINV 136
           C                                                                         MINV 137
 0065                A(KK)=1.0/BIGA                                                  MINV 138
 0066             80 CONTINUE                                                        MINV 139
           C                                                                         MINV 140
           C            FINAL ROW AND COLUMN INTERCHANGE                             MINV 141
           C                                                                         MINV 142
 0067                K=N                                                             MINV 143
 0068            100 K=(K-1)                                                         MINV 144
 0069                IF(K) 150,150,105                                               MINV 145
 0070            105 I=L(K)                                                          MINV 146
 0071                IF(I-K) 120,120,108                                             MINV 147
 0072            108 JQ=N*(K-1)                                                      MINV 148
 0073                JR=N*(I-1)                                                      MINV 149
 0074                DO 110 J=1,N                                                    MINV 150
 0075                JK=JQ+J                                                         MINV 151
 0076                HOLD=A(JK)                                                      MINV 152
 0077                JI=JR+J                                                         MINV 153
 0078                A(JK)=-A(JI)                                                    MINV 154
 0079            110 A(JI) =HOLD                                                     MINV 155
 0080            120 J=M(K)                                                          MINV 156
 0081                IF(J-K) 100,100,125                                             MINV 157
 0082            125 KI=K-N                                                          MINV 158
 0083                DO 130 I=1,N                                                    MINV 159
 0084                KI=KI+N                                                         MINV 160
 0085                HOLD=A(KI)                                                      MINV 161
 0086                JI=KI-K+J                                                       MINV 162
 0087                A(KI)=-A(JI)                                                    MINV 163
 0088            130 A(JI) =HOLD                                                     MINV 164
 0089                GO TO 100                                                       MINV 165
 0090            150 RETURN                                                          MINV 166
 0091                END                                                             MINV 167
```

```
FORTRAN IV G LEVEL  21                  LGRDRV            DATE = 76005          21/13/08

                SUBROUTINE LGRDRV(B,G,GDERIV)
                IMPLICIT REAL*8(A-H,O-$)
                COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
                DIMENSION B(15),BO(15),G(15),GO(15),GDERIV(225)
          C
          C   SUBROUTINE COMPUTES DERIVATIVES OF CONSTRAINTS W.R.T. PARAMETERS
          C
                DO 3 I=1,NPARM
                DO 1 J=1,NPARM
              1 BO(J)=B(J)
                H=B(I)*.00100
                H1=1.DO/H
                BO(I)=B(I)+H
                CALL LGRANG(BO,GO)
                M=(I-1)*NCONST
                DO 2 J=1,NCONST
              2 GDERIV(M+J)=(GO(J)-G(J))*H1
              3 CONTINUE
                RETURN
                END
```

```
FORTRAN IV G LEVEL  21                    TRANS                DATE = 76005           21/13/08

 0001                SUBROUTINE TRANS(B)
 0002                IMPLICIT REAL*8 (A-H,O-$)
 0003                COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
 0004                DIMENSION B(15)
 0005                RETURN
 0006                END
```

```
FORTRAN IV G LEVEL  21                 LGRANG              DATE = 76005           21/13/08

 0001                SUBROUTINE LGRANG(B,G)
 0002                IMPLICIT REAL*8(A-H,O-$)
 0003                COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N,NCONST
 0004                DIMENSICN B(15),G(15)
               C
               C   CCNSTRAINTS CN PARAMETERS ARE PLACED HERE
               C   WRITE CONSTRAINTS AS 0=CCNSTRAINTS. G(1) IS FIRST CCNSTRAINT, ETC...
               C
 0CC5                RETURN
 0C06                END
```

```
FORTRAN IV G LEVEL  21                  FUNC                DATE = 76005            21/13/08

                     SUBROUTINE FUNC(B,F)
                     IMPLICIT REAL*8(A-H,O-$)
                     COMMON X(1500),Y(100),DELTA(15),SS,NOBS,NVAR,NPARM,N
                     DIMENSION B(15),F(100)
              C
              C   FUNCTION TO BE MODELED IS SUPPLIED HERE
              C
                     DO 1 I=1,NOBS
                     IN1=I+NOBS
                     F(I)=DEXP((-B(1))*X(I)*DEXP((-B(2))/X(IN1)))
                   1 CONTINUE
                     RETURN
                     END
```

```
                    EXAMPLE PROBLEM FROM BARD, P.124

NONLINEAR ESTIMATION USING MARQUARDT PROCEDURE.

NO. OF OBS.=  15     NO. OF PARAMETERS=  2     NO. OF INDEP. VAR.=  2
MAXIMUM NO. OF ITERATIONS= 30  NO. OF USER-SUPPLIED CONSTRAINTS=  0

ORIGINAL PARAMETER ESTIMATES.
         1                   2
     750.00000          1200.00000

DATA AS INPUT TO PROGRAM NONLIN
             0.98000             0.10000            100.00000
             0.98300             0.20000            100.00000
             0.95500             0.30000            100.00000
             0.97900             0.40000            100.00000
             0.99300             0.50000            100.00000
             0.62600             0.05000            200.00000
             0.54400             0.10000            200.00000
             0.45500             0.15000            200.00000
             0.22500             0.20000            200.00000
             0.16700             0.25000            200.00000
             0.31700             0.04000            300.00000
             0.56600             0.02000            300.00000
             0.03400             0.06000            300.00000
             0.01600             0.08000            300.00000
             0.06600             0.10000            300.00000

INITIAL SUM OF SQUARES=                          1.090440905418776

                  ITERATION NO.  1
PARAMETER VALUES
         1                   2
     152.76696           703.14640
SUM OF SQUARES=                       0.448321270868956

                  ITERATION NO.  2
PARAMETER VALUES
         1                   2
     386.29234           928.50826
SUM OF SQUARES=                       0.409038973823016

                  ITERATION NO.  3
PARAMETER VALUES
         1                   2
     624.68082           928.70228
SUM OF SQUARES=                       0.055969223488254

                  ITERATION NO.  4
PARAMETER VALUES
         1                   2
     804.18994           965.48472
SUM OF SQUARES=                       0.040732717907191

                  ITERATION NO.  5
PARAMETER VALUES
         1                   2
     818.35389           962.09427
SUM OF SQUARES=                       0.039807476329395

                  ITERATION NO.  6
PARAMETER VALUES
         1                   2
         812.06041           960.51863
SUM OF SQUARES=                          0.039806236170666

              ITERATICN NO.  7
PARAMETER VALUES
         1                   2
         812.66423           960.65815
SUM OF SQUARES=                          0.039806135586079

              ITERATION NO.  8
PARAMETER VALUES
         1                   2
         812.66032           960.65862
SUM OF SQUARES=                          0.039806135503003

              ITERATION NO.  9
PARAMETER VALUES
         1                   2
         812.65999           960.65865
SUM OF SQUARES=                          0.039806135502997

ESTIMATION PROCEDURE TERMINATED DUE TO APPARENT CONVERGENCE TO THE LISTED PARAMETER VALUES.

INVERSE OF SS & CROSS PRODUCTS MATRIX OF PARTIALS EVALUATED AT FINAL B:

 0.19850D 08 0.54251D 07
 0.54251D 07 0.15397D 07
```

Acknowledgements

I wish to thank Dr. George Milliken for his technical assistance and moral support while preparing this report and throughout my graduate studies at Kansas State University; his open door has always been a welcome sight.

Gratitude is also extended to Drs. Dayton and Fryer, and the Department of Statistics for financial support during my studies.

Deepest thanks are reserved for my wife, Evelyn, who devotedly backed my every whimsical notion and decision. It has been her encouragement that has helped me strive for the self-development one hopes to gain by graduate study.

A STUDY OF THREE ALGORITHMS FOR NONLINEAR LEAST SQUARES PARAMETER ESTIMATION

by

MICKEY LINN STILSON

B.S., Kansas State University, 1971

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Statistics

KANSAS STATE UNIVERSITY

1976

Two analytic approaches toward minimizing the residual sum of squares function of the nonlinear model are presented. The first uses a first order Taylor series approximation of the nonlinear function; the second uses a gradient search. Both general techniques entail an iterative scheme for finding the least squares parameter estimate. From these two general approaches three currently used algorithms are developed. Examples are presented which show that no one algorithm is best for all problems. The topic of parameter estimation under parameter constraints is then discussed. An appendix documenting a computer program written by the author incorporating the algorithms and constrained estimation is provided.