GENERAL ALGORITHMS BASED ON LEAST SQUARES CALCULATIONS FOR MAXIMUM LIKELIHOOD ESTIMATION IN MULTIPARAMETER MODELS

A thesis presented in partial fulfilment of the requirements for the degree of Ph.D. in Statistics at Massey University.

by

W. Douglas Stirling

1985

ABSTRACT

This thesis develops algorithms for maximum likelihood estimation that can be implemented using a sequence of weighted least squares computations and examines their properties.

Standard least squares algorithms are first described and their execution times, storage requirements and accuracies are compared. The Givens QR algorithm uses less storage than other algorithms of comparable accuracy and in good implementations is virtually as fast as them if there are several explanatory variables. A version that can be used for constrained least squares is described; it is used for least squares calculations in the remainder of the thesis.

In many maximum likelihood problems, the likelihood can be written as a sum of functions called log-likelihood components and these often depend on the unknown parameters only through one or two quantities called systematic parts. For these models, a class of algorithms called NRL algorithms approaches the maximum likelihood estimate with a sequence of least squares calculations. For many common models, the Newton-Raphson algorithm and Fisher's scoring technique are particular NRL algorithms. Implementation of NRL algorithms is described in detail and the relative merits of the various NRL algorithms are discussed. If the NR algorithm is in the class, it converges best near the maximum likelihood estimate, but other NRL algorithms may perform better in the first few iterations. Several examples are analysed to illustrate the various possible methods.

When the maximum likelihood estimates of some parameters can be written as explicit functions of the rest, the convergence of the NR and NRL algorithms can often be improved by adjusting these parameters between iterations. The relationship of this technique to elimination of these parameters from the likelihood is investigated. In several types of model, including nonlinear least squares, adjustment can be

performed without slowing the NRL iterations. A related more general method is also described for improving NRL iterations when some parameters are linear and some are nonlinear in the systematic parts.

Another general algorithm called the EM algorithm is described. It can be applied to several types of model for which the NRL algorithm cannot be used. In some models, it can also be implemented using a sequence of least squares calculations, but for applications where both EM and NRL algorithms can be used, the latter usually converge faster.

Finally, in two appendices, Fortran subroutines that can be used to implement the algorithms in the thesis are described and listed.

ACKNOWLEDGEMENTS

I would like to thank Dr R J Brook for supervising this thesis. I am also grateful to the Department of Mathematics and Statistics for allowing me time to complete this thesis and to Massey University for making available computer facilities for typing it.

I finally wish to acknowledge the support my wife Sue has given me and to thank her and my children David and George for tolerating the evening and weekends I have spent on the thesis in the last few years.

TABLE OF CONTENTS

Appendices

1.INTRODUCTION

1.1 PURPOSE AND SCOPE OF THE THESIS

Most applications of statistics involve models for a vector of observed random variables $\mathbf{y}$ in which its distribution is specified, apart from the values of some unknown parameters , $\boldsymbol{\beta}$. Often maximum likelihood is used to estimate $\boldsymbol{\beta}$ from $\mathbf{y}$, largely because maximum likelihood estimators are easily defined, have desirable asymptotic properties and are used in the generalized likelihood ratio test.

The use of multiparameter models has been highly dependent on the ease with which they can be fitted. Early in the history of statistics, least squares algorithms were discovered for maximum likelihood parameter estimation in the normal linear model,

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}\sigma^2) \quad .$$

It is only more recently that numerical methods have been developed to apply maximum likelihood to other models and computers have become widely available to implement them at reasonable cost. However most of the effort of developing maximum likelihood algorithms has been directed at fairly narrow specific types of models and only a few papers, such as Nelder and Wedderburn (1972) and Dempster et al (1977), have developed general algorithms that can be applied to wide classes of models.

The aim of this thesis is to describe some general algorithms that can be easily used to efficiently find maximum likelihood estimates in a wide range of multiparameter statistical models. The algorithms examined in the thesis are all based on sequences of least squares calculations; they can be applied to most models that are commonly

used in statistics, the main exceptions being models that arise in multivariate analysis. For algorithms in such areas as cluster analysis, factor analysis, multidimensional scaling, etc., different types of algorithms are usually needed and the reader is referred for details to standard textbooks on multivariate analysis. Applications in multivariate analysis will not be considered in this thesis.

It is not claimed that the general algorithms described in the thesis will be the most efficient methods for all possible models. When they are illustrated by applying them to specific examples in the text, faster algorithms may exist. However the algorithms that we describe could be incorporated within a single computer package and that alone would make them attractive compared to alternative methods invented for specific models.

1.2 STRUCTURE OF THE THESIS

The problem of maximizing a log-likelihood function $\ell(\beta)$ is clearly a special case of unconstrained optimization. General optimization algorithms can therefore be used to find maximum likelihood estimates.

When applied to most types of model that are important in statistics, many general optimization algorithms can be implemented with a sequence of least squares calculations. In Chapter 2, least squares algorithms are therefore examined in detail, particularly with regard to storage requirements, speed and accuracy of evaluating fitted values.

In some models, the log-likelihood function $\ell(\beta)$ can be expressed in the form

$$\ell(\beta) = \sum \ell_i(\eta_i) + K$$

where $\eta_i = \eta_i(\beta)$ and K does not depend on β. In Chapter 3, algorithms based on least squares calculations are developed for models of this kind. Special cases which simplify the algorithms are defined and implementation is described in detail. Properties of the various algorithms of this type are compared. Variances of parameter estimates and tests of hypotheses about the parameters are discussed and the algorithms are illustrated with a few examples.

Chapter 4 extends the algorithms to models where

$$\ell(\beta) = \sum \ell_i(\eta_i^{(1)}, \eta_i^{(2)}, \ldots) + K$$

and where $\eta_i^{(1)} = \eta_i^{(1)}(\beta)$, $\eta_i^{(2)} = \eta_i^{(2)}(\beta)$, etc.. This framework covers most multiparameter models that are common in statistics. Special cases that lead to simplifications of the algorithms are identified. Several types of example are used to illustrate the algorithms.

Sometimes the maximum likelihood estimate of a subset of parameters can be expressed as an explicit function of the remaining parameters. This can be used either to eliminate the first subset from the likelihood function or to adjust the first subset between joint iterations for all parameters. In Chapter 5, these strategies are compared. They can both result in improvements of convergence. A major application is in nonlinear least squares; for that and some other types of model, adjustment can be added to iterations without increasing execution time per iteration.

A different type of algorithm that can be applied to some multiparameter models is an algorithm called the EM algorithm. The EM algorithm and its properties are described in Chapter 6 and some comparisons are made between it and the algorithms developed earlier in the thesis. The EM algorithm is mainly recommended for models that cannot be expressed in the form required for the algorithms in Chapters 3 and 4.

The methods in the thesis allow a wide class of multiparameter models to be fitted using a single framework which is based on a sequence of numerically stable least squares calculations. Implementing the algorithms in a single computer system would provide users with a system of considerable flexibility. The user interface of such a system is indicated in the conclusion.

2. LEAST SQUARES ALGORITHMS

2.1 THE NORMAL LINEAR MODEL

The ordinary normal linear model,

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}\sigma^2) \qquad (2.1.1)$$

and the associated method of estimating $\boldsymbol{\beta}$ by least squares, have been the most widely used statistical techniques in this century.

The first description of the method of least squares was given by Legendre in 1805 who treated the model

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + e_i \qquad i=1,\ldots n$$

where $e_1,\ldots,e_n$ were called "errors". He made no distributional assumptions but suggested estimating $\beta_1,\ldots,\beta_p$ to minimize

$$\sum e_i^2 = \sum_i (y_i - \sum_j x_{ij}\beta_j)^2$$

as an intuitively reasonable method.

Gauss in 1809 first introduced the concept of normally distributed errors and showed that with uniform priors on $\beta_1,\ldots,\beta_p$, the least squares estimates are at the mode of the posterior distribution of $\beta_1,\ldots,\beta_p$. At that time however normality was not regarded as important. Between 1821 and 1826 Gauss showed, using only assumptions of independence and constant variance, that the least squares estimators of $\beta_1,\ldots,\beta_p$ (and linear functions of these) have minimum variance amoung unbiased estimators that are linear in $y_1,\ldots,y_n$; this was used as the main justification for the method. It was only later, after the distributions of the residual sum of squares and the parameter estimates and tests about the parameters were derived and the

relationship of the model to the multivariate normal distribution was discovered, that normality became a standard assumption of the linear model and least squares. The matrix notation (2.1.1) was not used until Aitken(1935). Seal (1967) and Plackett (1972) discuss the historical development of least squares.

The model (2.1.1) is less restrictive than it might initially appear. Non-linear relationships between a response and an explanatory variable can be modelled by allowing transformations of either variable, or by adding extra "explanatory variables" which are powers of the original explanatory variable. A modification of this is to model the relationship with a different polynomial in different ranges of explanatory values, with constraints of continuity and smoothness of the polynomials (which are called splines) at the values of the explanatory variable where they meet (which are called knots); when the positions of these knots are fixed, the model can be again written in form (2.1.1). Seasonal trends can be incorporated with either seasonal factors or sine and cosine terms in time. The use of dummy 0/1 variables allows all fixed effect analysis of variance models to be expressed in the form (2.1.1). The normal model and least squares have therefore been applied to a wide variety of areas.

2.2 ALGORITHMS BASED ON THE NORMAL EQUATIONS

In the model $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}\sigma^2)$ which explains the distribution of a vector of n responses, $\mathbf{y}$, in terms of a vector of p unknown parameters, $\boldsymbol{\beta}$, and an unknown scalar parameter σ, the maximum likelihood least squares estimator of $\boldsymbol{\beta}$ is easily shown to satisfy the "normal" equations

$$(\mathbf{X}'\mathbf{X})\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{y} \qquad (2.2.1)$$

and this leads to the standard text-book formula

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \ .$$

($\mathbf{X}$ is assumed to be full rank; otherwise generalized inverses must be used).

Unfortunately, estimating $\boldsymbol{\beta}$ by solving the normal equations (2.2.1) is numerically unstable if the columns of $\mathbf{X}$ are multicollinear. This is easily seen when $E[y_i] = \beta_1 + \beta_2 x_i$ for $i=1,\ldots,n$. The least squares estimator of β_2 must be calculated from the normal equations as

$$\hat{\beta}_2 = \frac{\Sigma x_i y_i - n^{-1}(\Sigma x_i)(\Sigma y_i)}{\Sigma x_i^2 - n^{-1}(\Sigma x_i)^2}$$

If the coefficient of variation of $x_1,\ldots,x_n$ is small, there can be disastrous cancellation errors when evaluating the denominator. The mathematically equivalent formulae

$$\hat{\beta}_2 = \frac{\Sigma(x_i-\bar{x})y_i}{\Sigma(x_i-\bar{x})^2} \quad \text{or} \quad \hat{\beta}_2 = \frac{\Sigma(x_i-\bar{x})(y_i-\bar{y})}{\Sigma(x_i-\bar{x})^2}$$

provide an evaluation method that is much more stable numerically, but which does not use the coefficients in the normal equations.

By analogy with the single explanatory variable case, when there is a constant term in the model (which without loss of generality is

assumed to be the first element of $\boldsymbol{\beta}$), the normal equations (2.2.1) are often replaced by the equations

$$\begin{aligned} (\mathbf{X}^{*\prime}\mathbf{X}^{*})\hat{\boldsymbol{\beta}}_2 &= \mathbf{X}^{*\prime}\mathbf{y} \\ \hat{\beta}_1 + \hat{\boldsymbol{\beta}}_2'\bar{\mathbf{x}} &= \bar{y} \end{aligned} \qquad (2.2.2)$$

where $\bar{\mathbf{x}}$ is the vector of (p-1) means of the last (p-1) columns of $\mathbf{X}$, $\mathbf{X}^*$ is the matrix whose columns are these variables with their means subtracted, $\bar{y}$ is the mean of $\mathbf{y}$ and $\boldsymbol{\beta}' = [\beta_1 : \boldsymbol{\beta}_2']$. The sums of squares and products round the mean, $\mathbf{X}^{*\prime}\mathbf{X}^*$ and $\mathbf{X}^{*\prime}\mathbf{y}$, can be accumulated with either a 2-pass algorithm (the first pass through the data being to evaluate $\bar{\mathbf{x}}$ and $\bar{y}$) or using a stable 1-pass algorithm such as that described by Clarke(1977). Some computer packages find $\hat{\boldsymbol{\beta}}$ by solving equations (2.2.2).

Writing the equations (2.2.1) in the form (2.2.2), however, only avoids the numerical instability associated with one potential collinearity between the columns of $\mathbf{X}$. Any other collinearities can cause similar cancellation problems and can similarly result in numerically inaccurate estimates. In the next section, the idea in the algorithm implied by equations (2.2.2) is extended to provide more accurate estimates in the presence of any types of collinearity between the columns of $\mathbf{X}$. These algorithms do not use the coefficients of the normal equations, but are based on transformations of the columns of $\mathbf{X}$ and $\mathbf{y}$.

2.3 QR ALGORITHMS

The most accurate methods for finding least squares estimates are based on applying orthogonal transformations to the rows of the n×(p+1) matrix $[\mathbf{X} : \mathbf{y}]$ to reduce it to upper triangular form. These are called QR algorithms because of the corresponding factorization of $[\mathbf{X} : \mathbf{y}]$,

$$[\mathbf{X} : \mathbf{y}] = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \tag{2.3.1}$$

where $\mathbf{Q}$ is an n×n orthogonal matrix and $\mathbf{R}$ is (p+1)×(p+1) upper triangular. The matrix $\mathbf{R}$ in the factorization is the matrix that would be obtained from a Choleski factorization $[\mathbf{X} : \mathbf{y}]'[\mathbf{X} : \mathbf{y}] = \mathbf{R}'\mathbf{R}$; however the QR algorithms do not obtain $\mathbf{R}$ in this way. Note that the last (n-p-1) columns of $\mathbf{Q}$ are not unique. Once a factorization of this form has been determined, the least squares estimates can be easily calculated. If $\mathbf{R}$ is partitioned

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{r}_2 \\ \mathbf{0}' & r_3 \end{bmatrix}$$

where $\mathbf{R}_1$ is p×p upper triangular, then the normal equations (2.2.1) can be written as

$$\mathbf{R}_1'\mathbf{R}_1\hat{\boldsymbol{\beta}} = \mathbf{R}_1'\mathbf{r}_2 \quad .$$

When $\mathbf{X}$ is of full column rank, this equation can be rewritten as

$$\mathbf{R}_1\hat{\boldsymbol{\beta}} = \mathbf{r}_2 \tag{2.3.2}$$

which can be easily and accurately solved by back-substitution. Also r_3 is the square root of the residual sum of squares and the fitted values can be found from the formula

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{Q}_1\mathbf{r}_2 \tag{2.3.3}$$

where $\mathbf{Q}_1$ is the (unique) matrix consisting of the first p columns of $\mathbf{Q}$.

The three QR algorithms that will be described below differ in the way that $\mathbf{R}$ is built up, in the last (n-p-1) columns of $\mathbf{Q}$ (though these are not usually explicitly evaluated), and in the way that $\mathbf{Q}_1$ is represented if it is to be used later to find $\hat{\mathbf{y}}$ from (2.3.3).

(a) Modified Gram-Schmidt QR Algorithm

This method was suggested by Bjorck(1967). Using the notation $\mathbf{U}_i$ to denote the i'th column of $[\mathbf{X} : \mathbf{y}]$, the algorithm can be expressed as

```
for i from 1 to (p+1) do
  begin
    R_ii := √(U_i'U_i) ;
    U_i := U_i / R_ii ;
    for j from (i+1) to (p+1) do
        perform least squares fit of U_j against U_i;
        U_j := vector of residuals;
        R_ij := regression coefficient
  end
```

After the algorithm is finished, $\mathbf{X}$ is overwritten by $\mathbf{Q}_1$ and $\mathbf{y}$ is overwritten by $(\mathbf{y}-\hat{\mathbf{y}})$. (2.3.2) is then used to find $\hat{\boldsymbol{\beta}}$ from $\mathbf{R}$.

The main disadvantage with the algorithm is that storage for the whole data matrix $[\mathbf{X} : \mathbf{y}]$ is needed and this storage is not accessed sequentially. If n and p are large, an $n\times(p+1)$ array of fast random access memory is therefore required even when only $\mathbf{R}$ is wanted. Also the columns of $\mathbf{Q}_1$ may not be exactly orthogonal; this however does not generally cause problems for least squares calculations and, if necessary, the reorthogonalization method of Daniel et al (1976) may be used.

In the following two QR algorithms, a series of orthogonal transformations are applied to $[\mathbf{X} : \mathbf{y}]$ to reduce it to upper triangular form, so that

$$\mathbf{O}_k\mathbf{O}_{k-1} \cdots \mathbf{O}_2\mathbf{O}_1 \, [\mathbf{X} : \mathbf{y}] = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

where each multiplication by an orthogonal $\mathbf{O}_i$ introduces certain zeros. $\mathbf{Q} = \mathbf{O}_1'\mathbf{O}_2' \cdots \mathbf{O}_k'$ but this matrix rarely needs to be evaluated. Each $\mathbf{O}_i$ that introduces b zeros can be characterized by b values which can be stored instead of the zeros to allow the transformation to be reconstructed.

(b) Householder QR Algorithm

Householder transformations are pre-multiplications by orthogonal matrices of the form

$$\mathbf{O}_i = \mathbf{I} - \mathbf{v}_i\mathbf{v}_i'$$

where $\mathbf{v}_i'\mathbf{v}_i = 2$. Choice of $\mathbf{v}_i$ can be made such that

$$\mathbf{O}_i \, (\mathbf{O}_{i-1} \cdots \mathbf{O}_1 \, [\mathbf{X} : \mathbf{y}])$$

leaves the first (i-1) rows and columns of $(\mathbf{O}_{i-1} \cdots \mathbf{O}_1 \, [\mathbf{X} : \mathbf{y}])$ unaltered and makes its (j,i)'th elements zero for j=i+1,...,n . Then $(\mathbf{O}_{p+1} \cdots \mathbf{O}_1) = \mathbf{Q}'$. If $\mathbf{U}_i$ denotes the (n-i+1)-vector containing the last (n-i+1) elements in the i'th column of $\mathbf{O}_{i-1} \cdots \mathbf{O}_i \, [\mathbf{X} : \mathbf{y}]$, then $\mathbf{v}_i$ can be found from

$$S := \sqrt{(\mathbf{U}_i'\mathbf{U}_i)}$$
$$\text{if } U_{i1} > 0 \text{ then } S := -S$$
$$R_{ii} := S$$
$$U_{i1} := U_{i1} - S$$
$$\mathbf{v}_i := \begin{bmatrix} \mathbf{0} \\ \mathbf{U}_i \; / \; \text{sqrt}(\; -U_{i1} * S \;) \end{bmatrix}$$

Like the modified Gram-Schmidt algorithm, the Householder algorithm needs storage for the whole data matrix $[\mathbf{X} : \mathbf{y}]$ in fast random-access memory even if only $\mathbf{R}$ is needed.

(c) Givens QR Algorithm

Givens transformations are pre-multiplications by orthogonal matrices $\mathbf{O}_{ij}$ whose only non-zero off-diagonal elements are the (i,j)th and (j,i)th. Orthogonality implies that all diagonal elements are 1.0 except the (i,i)th and (j,j)th. The condition that the (i,j)th element of a particular matrix is transformed to zero by $\mathbf{O}_{ij}$ is enough to uniquely determine it. Givens transformations are usually applied to sequentially introduce zeros into $[\mathbf{X} : \mathbf{y}]$ by rows to make it upper triangular, so that

$$\mathbf{Q} = \prod_{i=2}^{n} \prod_{j=1}^{\min(i-1,p+1)} \mathbf{O}'_{ij} \quad .$$

Its main advantage over the Gram-Schmidt and Householder algorithms is that the whole data matrix $[\mathbf{X} : \mathbf{y}]$ does not need to be stored since the only rows altered when the i'th row is zeroed are the first p rows and the ith row; rows of $[\mathbf{X} : \mathbf{y}]$ can be considered one at a time then zeroed by Givens transformations. This is particularly useful when there are factors in a model whose levels do not then need to be stored in the expanded form of dummy variables; similarly, with polynomial terms in a model all powers do not need to be stored. (If the matrix $\mathbf{Q}_1$ must be reconstructed, an $n \times (p+1)$ matrix is however needed to keep track of the transformations used. However its elements are sequentially accessed and so do not need to be in random access memory and, at any rate, $\mathbf{Q}_1$ is not usually required).

The efficient organization of the Givens method is not widely known. It is most easily described by considering the effect of a transformation $\mathbf{O}_{ji}$ which zeroes the (j,i)th element of a matrix $\mathbf{U}$. If the i'th and j'th rows of $\mathbf{U}$ are denoted by $\mathbf{u}_i'$ and $\mathbf{u}_j'$, they are transformed into

$$\begin{Bmatrix} \mathbf{u}_i^{+\prime} \\ \mathbf{u}_j^{+\prime} \end{Bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{Bmatrix} \mathbf{u}_i' \\ \mathbf{u}_j' \end{Bmatrix} = \begin{Bmatrix} c\mathbf{u}_i' + s\mathbf{u}_j' \\ c\mathbf{u}_j' - s\mathbf{u}_i' \end{Bmatrix}$$

where $c = u_{ii}/\sqrt{(u_{ii}^2 + u_{ji}^2)}$ and $s = u_{ji}/\sqrt{(u_{ii}^2 + u_{ji}^2)}$ and the remaining rows of $\mathbf{U}$ are unaltered. Direct application of this formula would

involve more than twice the number of multiplications that are used in the Gram-Schmidt and Householder methods.

The number of operations can however be reduced if a different representation is used for the rows of $\mathbf{U}$. If $\mathbf{u}_m = \sqrt{w_i}\mathbf{z}_i$ where w_i can be thought of as a weight for the row $\mathbf{z}_i$ and a similar representation is used for $\mathbf{u}_i^+$ $(i=1,\ldots,n)$, then the transformation can be expressed as

$$\begin{bmatrix}\mathbf{z}_i^{+\prime}\\ \mathbf{z}_j^{+\prime}\end{bmatrix} = \begin{bmatrix}c\sqrt{(w_i/w_i^+)}\mathbf{z}_i' + s\sqrt{(w_j/w_i^+)}\mathbf{z}_j'\\ c\sqrt{(w_j/w_j^+)}\mathbf{z}_j' - s\sqrt{(w_i/w_j^+)}\mathbf{z}_i'\end{bmatrix} \qquad (2.3.4)$$

where w_i^+ and w_j^+ can be arbitrarily chosen. This

implies that, on input, $[\mathbf{X} : \mathbf{y}]$ should be represented by $\mathbf{W}$ and $[\mathbf{X}^* : \mathbf{y}^*]$ where $\mathbf{W}$ is the diagonal matrix whose diagonal elements are the weights of the rows of $[\mathbf{X}^* : \mathbf{y}^*]$ and $[\mathbf{X} : \mathbf{y}] = \mathbf{W}^{1/2}[\mathbf{X}^* : \mathbf{y}^*]$. This is particularly convenient for weighted least squares problems; for unweighted problems, $\mathbf{W} = \mathbf{I}$. After all Givens rotations have been completed to reduce $[\mathbf{X} : \mathbf{y}]$ to upper triangular, $\mathbf{R}$ is represented by $\mathbf{D}$ and $\mathbf{R}^*$ where $\mathbf{R} = \mathbf{D}^{1/2}\mathbf{R}^*$, $\mathbf{R}^*$ is upper triangular and $\mathbf{D}$ is a diagonal matrix of weights for the rows of $\mathbf{R}^*$. To evaluate $\hat{\boldsymbol{\beta}}$, we can solve

$$\mathbf{R}_1^*\hat{\boldsymbol{\beta}} = \mathbf{r}_2^*$$

instead of (2.3.2) since the weights $\mathbf{D}$ do not affect the solution to (2.3.2).

Gentleman(1973) suggested $w_j^+ = w_j c^2$ and $w_i^+ = (w_i z_{ii}^2 + w_j z_{ji}^2)$ in (2.3.4) which makes one of the coefficients of the vectors in the right hand side of (2.3.4) become unity and reduces the number of multiplications by approximately a quarter; it also gives $z_{ii}^+ = 1$ (so that $\mathbf{R}^*$ becomes unit upper triangular). Hammarling(1974) and Wilkinson(1977) suggested $(w_i^+ = c^2 w_i\ ,\ w_j^+ = c^2 w_j)$ if $c>s$ or $(w_i^+ = s^2 w_j\ ,\ w_j^+ = s^2 w_i)$ if $c\leq s$ which makes two coefficients on the right hand side of (2.3.4) unity and approximately halves the multiplications ; this results in an algorithm comparable in speed with the Householder and Gram-Schmidt algorithms. Since the weights may be reduced by up to a half in each of Hammarling's Givens

rotations, there is some danger of underflow; however this can be avoided by periodic rescaling of rows, which can be done with integer arithmetic on exponents and is therefore fast.

TABLE 2.3.1

Floating point multiplications and square roots needed to find upper triangular **R**, where $[\mathbf{X} : \mathbf{y}]'[\mathbf{X} : \mathbf{y}] = \mathbf{R}'\mathbf{R}$

	First q rows†		Each additional row	
	Multiplications	Square roots	Multiplications	Square roots
Choleski	$\frac{2}{3}q^3 + q^2 + O(q)$	q	$\frac{q(q+1)}{2}$	0
Modified Gram-Schmidt	$q^3 + \frac{1}{2}q^2 + O(q)$	q	q^2	0
Householder	$\frac{2}{3}q^3 + \frac{1}{2}q^2 + O(q)$	q	q^2	0
Gentleman's Givens	$q^3 + \frac{3}{2}q^2 + O(q)$	0	$\frac{3}{2}q^2 + \frac{9}{2}q - 3$	0
Hammarling's Givens	$\frac{2}{3}q^3 + 3q^2 + O(q)$	0	$q^2 + 7q - 2$	0

† q denotes the number of columns in $[\mathbf{X} : \mathbf{y}]$

We next examine the execution times of the Gram-Schmidt and Householder algorithms, the Gentleman and Hammarling versions of the Givens algorithm and an algorithm based on the normal equations that finds **R** from a Choleski factorization of $[\mathbf{X} : \mathbf{y}]'[\mathbf{X} : \mathbf{y}]$. Table 2.3.1 gives formulae for the numbers of floating point multiplications and square roots for determination of **R** using efficiently written versions of the five algorithms; the total operations for the first (p+1) rows

of $[\mathbf{X} : \mathbf{y}]$ is reported and also the operations per extra row thereafter. It is assumed that $[\mathbf{X} : \mathbf{y}]$ is not sparse and that it is of full column rank. For convenience we denote the number of columns of $[\mathbf{X} : \mathbf{y}]$ by $q=p+1$. The numbers of floating point additions are approximately the same as the numbers of floating point multiplications and are not reported in the table.

In problems with large p and n not much larger than p, the Householder algorithm, Hammarling's Givens algorithm and the Choleski algorithm are fastest since the execution times are dominated by operations in the first q rows. In problems with p large and $n >> p$, the execution times are dominated by operations in rows after the first q and so the Choleski algorithm based on the normal equations is fastest and uses half the floating point multiplications of the Gram-Schmidt algorithm, the Householder algorithm and Hammarling's Givens algorithm; Gentleman's Givens algorithm uses three times the floating point multiplications of the Choleski algorithm. In all types of problems with large p, the Householder algorithm and Hammarling's Givens algorithm are therefore comparably fast; though they are slower than the Choleski algorithm for $n >> p$, they are numerically more accurate and so would be preferred. Hammarling's Givens algorithm has several advantages over the Householder algorithm :-

(i) Most of the uses of least squares algorithms in later sections are for weighted problems. The two Givens algorithms take no longer for weighted least squares problems, whereas the other algorithms need q extra multiplications and a square root per row of $[\mathbf{X} : \mathbf{y}]$.

(ii) If $[\mathbf{X} : \mathbf{y}]$ is sparse, the operations in the Givens algorithm are often substantially reduced from the numbers in Table 2.3.1.

(iii) As mentioned above, when only $\mathbf{R}$ is needed, the Householder and Gram-Schmidt algorithms have considerably greater storage requirements. This is particularly important if $n >> p$.

However in problems with large n and small p, the Householder algorithm is somewhat faster.

In view of these points, Hammarling's Givens algorithm seems the most satisfactory of the QR algorithms for general use as regards speed and storage, despite its relative complication. Accuracy of all types of algorithms will be compared in the next section.

2.4 A NUMERICAL COMPARISON OF THE ACCURACY OF THE ALGORITHMS

In this section, we investigate the numerical accuracy of the modified Gram-Schmidt (MGS), Householder (HOUS) and Givens (GIV) algorithms described in Section 2.3 and one algorithm (CHOL) based on the normal equations which finds the factor $\mathbf{R}$ in (2.3.1) from a Choleski factorization of $[\mathbf{X} : \mathbf{y}]'[\mathbf{X} : \mathbf{y}]$. The versions of these algorithms used in the investigation were not optimized for execution speed in any way and no attempt was made to avoid square roots. Modifications to improve execution time need not however affect accuracy in any substantial way.

Our empirical comparison is based on applying the algorithms to a series of problems based on ones that were used by Wampler (1970) for a study of the accuracy of least squares algorithms in computer packages. The problems are least squares fits of polynomials of degree 4 and 5 in a variate X to 21 "responses" which were generated from the equation

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4 \ (+\beta_5 X^5) + \alpha E$$

where the 21 values of X and E are

X		E
0		759
1		-2048
2		2048
3		-2048
4		2523
5		-2048
6		2048
7		-2048
8		1838
9		-2048
10	and	2048
11		-2048
12		1838
13		-2048
14		2048
15		-2048
16		2523
17		-2048
18		2048
19		-2048
20		759

respectively. Four scales were used for the errors,

E1 : $\alpha = 0$
E2 : $\alpha = 1$
E3 : $\alpha = 100$
E4 : $\alpha = 10{,}000$

and for each degree of polynomial, four sets of parameters $\{\beta_i\}$ were used, as shown in Table 2.4.1. The coefficients are such that Y can be exactly stored in single precision on the Prime 750 computer that was used (which was not true with some of Wampler's examples). Since the errors are orthogonal to the explanatory variables in all the least squares problems, the least squares estimates of the parameters should be the known coefficients, and this allows the accuracy of the computed solutions obtained by the various algorithms to be assessed.

TABLE 2.4.1
Parameters used for generation of artificial polynomial regressions

	β_0	β_1	β_2	β_3	β_4	β_5
Degree 4						
P1 :	1	1	1	1	1	
P2 :	10000	1000	100	10	1	
P3 :	10000	-1000	100	-10	1	
P4 :	1634	-2120	506	-40	1	
Degree 5						
P5 :	1	1	1	1	1	1
P6 :	100000	10000	1000	100	10	1
P7 :	100000	-10000	1000	-100	10	-1
P8 :	7230	-16523	6370	-879	50	-1

For each algorithm and each least squares problem, the average number of correct decimal digits in the parameter estimates, $\bar{C}$, was calculated as the average of values

$$C_i = \begin{cases} -\log_{10} \left| \dfrac{\beta_i - \hat{\beta}_i}{\beta_i} \right| & \text{if } \beta_i \neq \hat{\beta}_i \\ \text{the approximate no of decimal digits to which the machine computes} & \text{if } \beta_i = \hat{\beta}_i \end{cases}$$

These are given in Tables 2.4.2 and 2.4.3 for both single precision and double precision versions of the algorithms. The three types of QR algorithm evaluate $\hat{\boldsymbol{\beta}}$ with comparable accuracy, confirming the claims in Chambers (1977, page 120) and Wilkinson (1977, section 5). The Choleski algorithm based on the normal equations was considerably less accurate. However moving from single to double precision had considerably more effect on accuracy than any differences between the algorithms. The other main feature of the tables is that the accuracy of all algorithms decreases when the errors get bigger.

In later chapters where least squares algorithms are used, we shall often be interested in determination of fitted values, rather than parameter estimates. We next compare various least squares algorithms for the accuracy of their determination of fitted values in the polynomial problems used above.

Each of the QR algorithms can obtain the fitted values in two ways as either $\hat{\mathbf{y}} = \mathbf{X}\mathbf{R}_1^{-1}\mathbf{r}_2$ or $\hat{\mathbf{y}} = \mathbf{Q}_1\mathbf{r}_2$ where $\mathbf{Q}_1$, $\mathbf{R}_1$ and $\mathbf{r}_2$ are defined in Section 2.3. We denote these two versions of the modified Gram-Schmidt algorithm by MGS-X and MGS-Q, with a similar notation for the Householder and Givens algorithms. We are in particular interested in the circumstances in which the formula $\hat{\mathbf{y}} = \mathbf{X}\mathbf{R}_1^{-1}\mathbf{r}_2$ performs well; if it performs badly compared to $\hat{\mathbf{y}} = \mathbf{Q}_1\mathbf{r}_2$ then a representation of $\mathbf{Q}_1$ must be stored by the algorithms. Even though this storage would be accessed sequentially for the Givens algorithm, the overhead would destroy an important aspect of the advantage of the Givens algorithm over the other QR algorithms.

For each polynomial problem and each algorithm, the average number of correct decimal digits in the fitted values was obtained as

TABLE 2.4.2

Number of correct decimal digits in computed parameter estimates for polynomial problems of degree 4

Single precision (6.9 significant decimal digits)

	P1				P2			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-X	3.44	2.92	1.29	-0.70	5.34	5.23	3.28	1.29
HOUS-X	3.21	3.14	0.98	-1.01	4.59	5.59	2.98	0.99
GIV-X	4.05	3.00	1.40	-0.63	5.24	4.83	3.41	1.37
CHOL-X	0.77	1.49	0.88	-0.64	2.35	2.14	3.28	1.98

	P3				P4			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-X	5.88	5.25	3.29	1.30	5.82	5.72	3.46	1.48
HOUS-X	4.79	5.36	2.98	1.00	5.91	5.20	3.14	1.17
GIV-X	5.33	5.02	3.41	1.39	5.00	5.21	3.56	1.55
CHOL-X	3.00	3.00	2.76	1.20	3.09	3.09	3.00	1.53

Double precision (14.1 significant decimal digits)

	P1				P2			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-X	10.65	10.79	8.70	6.67	12.60	12.45	10.70	8.67
HOUS-X	11.01	10.31	8.65	7.10	11.89	11.61	10.60	9.14
GIV-X	11.17	10.29	8.50	6.53	13.88	12.00	10.51	8.53
CHOL-X	7.81	7.81	7.81	7.81	9.81	9.81	9.81	9.81

	P3				P4			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-X	12.42	13.05	10.70	8.61	12.81	13.27	10.87	8.78
HOUS-X	11.91	11.93	11.79	8.57	12.08	12.06	11.15	8.78
GIV-X	12.54	12.71	10.51	8.52	12.20	12.33	10.68	8.67
CHOL-X	11.38	11.47	11.47	11.47	10.98	10.98	10.98	10.98

$$\bar{D} = \frac{\Sigma \left| \hat{Y} - Y + \alpha E \right|}{\Sigma \left| Y - \alpha E \right|}$$

These values are shown in Tables 2.4.4 and 2.4.5. There is not much difference between the accuracies of the fitted values using the two formulae for $\hat{\mathbf{y}}$. The formula $\hat{\mathbf{y}} = \mathbf{X}\mathbf{R}_1^{-1}\mathbf{r}_2$ gave slightly more accurate results if $\sum \beta_i X^i$ could be calculated without large cancellation errors (sets of parameters denoted by P1, P2, P3, P5, P6 and P7), whereas the formula $\hat{\mathbf{y}} = \mathbf{Q}_1\mathbf{r}_2$ was slightly more accurate if $\sum \beta_i X^i$ involved cancellation errors (sets of parameters P4 and P8). From this result, the version of the Givens algorithm that does not store $\mathbf{Q}_1$ and evaluates $\hat{\mathbf{y}}$ from the formula $\hat{\mathbf{y}} = \mathbf{X}\mathbf{R}_1^{-1}\mathbf{r}_2$ appears satisfactory for most problems and the use of double precision real variables is the best way to get more accuracy in the fitted values.

In this section it should finally be mentioned that the use of accurate QR algorithms does <u>not</u> mean that no thought needs to be given by the user to multicollinearity in his problem. A prior linear transformation of the variables should always be done by the user to reduce the multicollinearity <u>if that can be done without loss of significant digits</u>. For example in polynomial regression with equally spaced explanatory variables, the use of orthogonal polynomials gives more accurate answers than the most accurate algorithm applied to ordinary powers of the explanatory variables. In many other examples, a rough prior centering of the data helps.

TABLE 2.4.3

Number of correct decimal digits in computed parameter estimates for polynomial problems of degree 5

Single precision (6.9 significant decimal digits)

	P1				P2			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-X	2.92	2.80	0.80	-1.20	5.15	4.93	3.32	1.29
HOUS-X	2.71	1.70	0.04	-1.96	3.92	4.32	2.55	0.54
GIV-X	2.44	2.46	0.89	-1.10	4.69	4.83	3.36	1.39
CHOL-X	-1.02	-1.02	-1.02	-0.98	1.09	1.64	1.64	1.41

	P3				P4			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-X	4.81	5.26	3.31	1.30	5.17	5.18	3.54	1.55
HOUS-X	4.42	5.35	2.54	0.53	4.38	4.85	2.81	0.79
GIV-X	4.57	4.66	3.43	1.40	4.40	4.43	4.03	1.66
CHOL-X	1.15	1.15	1.15	1.08	1.22	1.22	1.22	1.22

Double precision (14.1 significant decimal digits)

	P1				P2			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-X	9.72	10.03	8.18	6.02	12.27	11.62	10.53	8.52
HOUS-X	9.41	9.86	7.82	5.80	11.40	11.78	10.37	8.94
GIV-X	9.97	9.79	7.76	5.76	12.28	11.78	10.26	8.26
CHOL-X	6.54	6.54	6.54	6.54	8.13	8.12	8.12	8.12

	P3				P4			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-X	11.99	12.09	10.67	8.66	11.99	12.01	10.90	8.91
HOUS-X	11.28	11.25	10.26	8.94	11.74	11.52	10.49	8.56
GIV-X	12.03	12.39	10.27	8.26	12.12	11.91	10.52	8.51
CHOL-X	8.22	8.32	8.33	8.32	8.06	8.06	8.06	8.06

TABLE 2.4.4

Number of correct decimal digits in computed fitted values for polynomial problems of degree 4

Single precision (6.9 significant decimal digits)

	P1				P2			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-Q	6.37	6.56	5.15	3.16	6.28	6.30	5.53	3.53
MGS-X	6.65	6.86	5.16	3.16	6.51	6.68	5.53	3.53
HOUS-Q	5.57	5.56	4.79	2.81	5.71	5.67	5.15	3.19
HOUS-X	6.53	6.54	4.79	2.81	6.42	6.39	5.17	3.19
GIV-Q	5.83	5.85	5.23	3.25	5.72	5.75	5.54	3.64
GIV-X	6.64	6.77	5.27	3.25	6.40	6.48	5.66	3.64
CHOL-X	4.66	5.35	4.73	3.26	4.58	4.39	5.20	4.21

	P3				P4			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-Q	6.20	6.26	5.02	3.03	4.89	4.88	3.47	1.49
MGS-X	6.55	6.41	5.02	3.03	5.28	4.99	3.47	1.49
HOUS-Q	5.55	5.58	4.64	2.68	4.93	4.85	3.10	1.13
HOUS-X	6.13	5.88	4.64	2.68	4.34	4.49	3.10	1.13
GIV-Q	5.77	5.73	5.12	3.13	5.15	5.11	3.57	1.58
GIV-X	6.01	5.92	5.14	3.13	4.25	4.17	3.57	1.58
CHOL-X	4.76	4.76	4.50	2.95	3.12	3.11	3.02	1.57

Double precision (14.1 significant decimal digits)

	P1				P2			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-Q	13.47	13.52	12.56	10.47	13.74	13.65	12.95	10.93
MGS-X	13.55	13.71	12.56	10.54	13.95	13.67	12.97	10.93
HOUS-Q	12.81	12.80	12.60	10.49	12.79	12.84	12.74	10.92
HOUS-X	13.78	13.84	12.44	10.49	13.42	13.32	12.82	10.92
GIV-Q	13.10	13.03	12.36	10.41	13.17	13.07	12.72	10.79
GIV-X	14.14	13.78	12.38	10.41	15.51	13.75	12.77	10.79
CHOL-X	11.69	11.69	11.69	11.69	12.07	12.07	12.07	12.07

	P3				P4			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-Q	13.47	13.50	12.37	10.42	12.04	12.04	10.81	8.88
MGS-X	13.42	13.40	12.43	10.35	11.80	11.66	10.88	8.80
HOUS-Q	12.70	12.72	12.35	10.30	12.16	12.11	10.81	8.78
HOUS-X	13.10	13.24	12.60	10.30	11.46	11.48	10.74	8.78
GIV-Q	13.12	13.00	12.24	10.26	12.36	12.38	10.70	8.70
GIV-X	13.32	13.16	12.25	10.26	11.77	11.77	10.70	8.70
CHOL-X	12.47	12.46	12.46	12.46	10.42	10.42	10.42	10.42

TABLE 2.4.5

Number of correct decimal digits in computed fitted values for polynomial problems of degree 5

Single precision (6.9 significant decimal digits)

	P1				P2			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-Q	6.22	6.15	5.77	3.77	6.28	6.30	6.06	4.14
MGS-X	6.64	6.63	5.78	3.77	6.76	6.70	6.15	4.15
HOUS-Q	5.41	5.39	4.97	2.99	5.55	5.54	5.27	3.36
HOUS-X	6.34	6.67	5.00	2.99	6.33	6.40	5.37	3.36
GIV-Q	5.85	5.84	5.63	3.89	5.76	5.82	5.78	4.25
GIV-X	6.59	6.71	5.87	3.89	6.54	6.88	6.23	4.25
CHOL-X	3.93	3.93	3.93	4.00	3.94	4.38	4.38	4.29

	P3				P4			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-Q	5.79	5.76	5.46	3.56	4.28	4.28	3.60	1.60
MGS-X	5.90	5.86	5.49	3.56	3.96	4.22	3.62	1.60
HOUS-Q	5.70	5.55	4.75	2.77	4.20	4.21	2.81	0.81
HOUS-X	5.80	5.88	4.77	2.77	4.08	4.05	2.82	0.81
GIV-Q	6.30	6.35	5.66	3.67	4.46	4.47	3.70	1.71
GIV-X	5.81	5.77	5.59	3.67	3.74	3.61	3.51	1.71
CHOL-X	3.40	3.40	3.40	3.32	1.23	1.23	1.23	1.24

Double precision (14.1 significant decimal digits)

	P1				P2			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-Q	13.51	13.47	12.97	11.13	13.51	13.75	13.33	11.50
MGS-X	13.79	13.67	13.12	10.99	13.82	14.08	13.34	11.37
HOUS-Q	12.73	12.67	12.63	10.86	12.73	12.71	12.74	11.73
HOUS-X	13.86	13.66	12.77	10.75	13.48	13.54	13.20	11.73
GIV-Q	13.06	13.19	12.72	10.72	13.11	13.11	12.98	11.09
GIV-X	13.66	13.95	12.72	10.72	14.16	14.03	13.08	11.09
CHOL-X	11.42	11.42	11.42	11.42	10.96	10.96	10.96	10.96

	P3				P4			
	E1	E2	E3	E4	E1	E2	E3	E4
MGS-Q	12.96	12.97	12.78	10.92	11.14	11.14	10.91	8.95
MGS-X	13.04	13.03	12.80	10.92	11.05	11.06	10.83	8.95
HOUS-Q	12.70	12.68	12.65	11.14	11.41	11.33	10.57	8.58
HOUS-X	13.02	12.97	12.50	11.14	10.73	10.89	10.50	8.58
GIV-Q	13.43	13.51	12.51	10.50	11.90	11.86	10.54	8.54
GIV-X	13.12	13.08	12.51	10.50	10.99	11.11	10.51	8.54
CHOL-X	10.46	10.56	10.57	10.56	8.08	8.08	8.08	8.08

2.5 LINEAR CONSTRAINTS

We next consider the problem of weighted least squares subject to q linear constraints on the parameters,

$$\text{minimize } (\mathbf{y}-\mathbf{X}'\boldsymbol{\beta})'\mathbf{W}(\mathbf{y}-\mathbf{X}'\boldsymbol{\beta}) \text{ subject to } \mathbf{C}\boldsymbol{\beta}=\mathbf{c}$$

where $\mathbf{C}$ and $\mathbf{c}$ are a matrix and vector respectively, of known constants and $\mathbf{W}$ is a diagonal matrix whose diagonal elements are known weights.

Theoretically the problem can be solved by using the constraints to eliminate parameters in $\boldsymbol{\beta}$, thereby reducing the problem to unconstrained least squares. In practice, however, this method often leads to numerical instability. Another more stable method is described by Golub(1965). Using Lagrange multipliers $\boldsymbol{\lambda}$, the solution can be shown to be $\hat{\boldsymbol{\beta}}$ found by successively solving

(a) $\mathbf{X}'\mathbf{W}\mathbf{X}\mathbf{b} = \mathbf{X}'\mathbf{W}\mathbf{y}$ for $\mathbf{b}$

(b) $\mathbf{C}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-}\mathbf{C}'\boldsymbol{\lambda} = \mathbf{C}\mathbf{b}-\mathbf{c}$ for $\boldsymbol{\lambda}$, where $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-}$ is a generalized inverse of $(\mathbf{X}'\mathbf{W}\mathbf{X})$

(c) $\mathbf{X}'\mathbf{W}\mathbf{X}\boldsymbol{\delta}+\mathbf{C}'\boldsymbol{\lambda} = \mathbf{0}$ for $\boldsymbol{\delta}$

(d) $\hat{\boldsymbol{\beta}} = \mathbf{b}+\boldsymbol{\delta}$

Stirling(1981) however described a much simpler algorithm, which is also numerically stable. If $\mathbf{c}$ is considered to be a prior estimate of $\mathbf{C}\boldsymbol{\beta}$ that is independent of $\mathbf{y}$ and has variance-covariance matrix $\text{var}(\mathbf{c}) = (\sigma^2/k)\mathbf{I}$, then the maximum likelihood estimate of $\boldsymbol{\beta}$ (assuming normality) is the weighted least squares solution using explanatory and response variables,

$$\mathbf{X}^* = \begin{Bmatrix} \mathbf{X} \\ \mathbf{C} \end{Bmatrix} \quad \text{and} \quad \mathbf{y}^* = \begin{Bmatrix} \mathbf{y} \\ \mathbf{c} \end{Bmatrix},$$

and vector of weights

$$\mathbf{w}^* = \begin{Bmatrix} \mathbf{w} \\ k\mathbf{e}_q \end{Bmatrix},$$

where $\mathbf{e}_q$ is a vector of q ones and $\mathbf{w}$ is the diagonal of $\mathbf{W}$. The prior estimates are therefore included as pseudo-observations with weight k.

It is intuitively obvious that the exact constraints can be imposed by letting $k \to \infty$. At $k=\infty$ the estimate, $\hat{\boldsymbol{\beta}}$, can be shown to satisfy $\mathbf{C}\hat{\boldsymbol{\beta}} = \mathbf{c}$ and the weighted residual sum of squares from $\mathbf{X}^*$, $\mathbf{y}^*$ and $\mathbf{w}^*$ can be shown to be $(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})'\mathbf{W}(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})$; $\hat{\boldsymbol{\beta}}$ therefore minimizes $(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})'\mathbf{W}(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})$ subject to $\mathbf{C}\hat{\boldsymbol{\beta}} = \mathbf{c}$.

The constrained least squares solution can therefore be found by adding constraints as extra observations with infinite weights, provided a least squares algorithm can be modified to take such infinite weights. The Givens least squares algorithms described earlier can be easily modified to accept such infinite weights if a variance $v_i = w_i^{-1}$ is stored for each row of data and each row of $\mathbf{R}$ rather than weight w_i. Constraints can be imposed by including the corresponding row with $v_i = 0$. Stirling(1981) showed that the updating formulae (2.3.4) for Gentleman's Givens algorithm using variances are as simple as those with weights. This is also true for Hammarling's Givens algorithm.

2.6 DEPENDENCIES BETWEEN EXPLANATORY VARIABLES

Up to here it has been assumed either that $\mathbf{X}$ is of full column rank or, if there are linear relationships between the columns of $\mathbf{X}$, that identifiability constraints have been imposed to define a unique $\hat{\boldsymbol{\beta}}$. In theory, if there is an unexpected random linear dependency between the columns of $\mathbf{X}$, there will be a zero diagonal element of $\mathbf{R}$ (r_{ii}^2 is the residual sum of squares from regressing the i'th variable against the first (i-1) variables). The correct least squares answer should be that the parameter estimates corresponding to the columns of $\mathbf{X}$ involved in the linear dependency are not unique and therefore have infinite variances and correlations of $\pm$ 1.0. In practice however, rounding errors often result in small non-zero diagonal elements of $\mathbf{R}$. Because of the rounding errors, the algorithms report that there are unique estimates with very large variances and correlations. These "unique" estimates are themselves often very large in magnitude and there can be large cancellation erors when finding fitted values and residuals from them. These and the residual sum of squares are numerically unstable and cannot be used as measures of the fit of the model.

Unfortunately there is no satisfactory way to numerically detect singularity and distinguish it from near-singularity. Because of lack of symmetry of the inclusion of the explanatory variables in $\mathbf{R}$, the absolute values of the diagonal elements of $\mathbf{R}$ do not themselves provide a good criterion to decide on whether a singularity may exist. The information in $\mathbf{R}$ about the multicollinearity of $\mathbf{X}$ is more easily interpreted if further orthogonal transformations are applied from the left and right to reduce $\mathbf{R}$ to diagonal form, say

$$\mathbf{Q}^*\mathbf{R}\,\mathbf{P}^* = \mathbf{D} \quad . \tag{2.6.1}$$

This and (2.3.1) lead to the factorization $\mathbf{X} = (\mathbf{Q}_1\mathbf{Q}^{*\prime})\mathbf{D}\,\mathbf{P}^{*\prime}$ which is called the singular value decomposition (SVD) of $\mathbf{X}$ and the diagonal elements of $\mathbf{D}$ are the singular values of $\mathbf{X}$. Golub and Reinch(1970)

described an iterative algorithm to find the factors in (2.6.1). It is desirable to scale the columns of $\mathbf{X}$ so that each has zero mean and unit sum of squares before a SVD is attempted since the squares of the singular values are then the eigenvalues of the correlation matrix of the explanatory variables and are easier to interpret as principal components.

The size of the smallest singular value seems to give the most easily interpreted measure of whether there might be a singularity in $\mathbf{X}$ and the corresponding column of $\mathbf{P}^*$ gives the (near-)linear relationship between the columns of $\mathbf{X}$. However because of the difficulty of finding the SVD, it is not routinely used in least squares calculations. An upper bound on the condition number (ratio of largest to smallest singular value) can however be found from the upper triangular matrix $\mathbf{R}$ (Karasalo(1974), Anderson and Karasalo(1975) and Lemeire(1975)). If the user can specify the value of the condition number above which he will accept that a singularity has occurred, the SVD need only be evaluated when the upper bound from $\mathbf{R}$ is above this value.

The final decision about whether or not to assume that a singularity has occurred and that an identifiability constraint is needed, is however still subjective. If a singularity is accepted, the most satisfactory solution is for the user to specify an identifiability constraint, aided by the columns of $\mathbf{P}^*$. If the user is unwilling or unable to do this, it is possible to find the unique solution $\hat{\boldsymbol{\beta}}$ with minimum length (Kennedy and Gentle, 1980, pages 318-319).

2.7 A SET OF FORTRAN SUBROUTINES FOR LEAST SQUARES

The least squares calculations that are performed in later chapters of this thesis are all done using the Fortran subroutines in Appendix B. Appendix A gives details of the parameters for each subroutine.

Subroutines GIVEN3 and GIVEN2 in Appendix B perform Gentleman's(1973) and Hammarling's(1974) versions, respectively, of the Givens algorithm to incorporate an extra row of data into an upper triangular matrix. Both subroutines use variances for each row rather than weights and can therefore be used with constraints as described in Section 2.5. Subroutine GIVEN3 was published by Stirling(1981). Gentleman's Givens algorithm ensures that the diagonal elements of the upper triangular matrix remain one and therefore is able to use a more compact storage arrangement than Hammarling's version. Subroutine CONV23 converts from the GIVEN2 unpacked format to the GIVEN3 packed format and subroutine INITG returns the representation of the zero matrix in packed format.

Two further subroutines are provided, both of which are minor changes of subroutines published by Stirling(1981). BSUB solves (2.3.2) by back-substitution from the GIVEN3 packed representation of $\mathbf{R}$. Subroutine VARS evaluates $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1} = \mathbf{R}^{-1}\mathbf{R}^{-1}{}'$, which is proportional to the variance-covariance matrix of the least squares estimates. If there are p parameters in $\boldsymbol{\beta}$, the residual sum of squares is the $(p+1)(p+2)/2$'th element in the packed representation of $\mathbf{R}$, but a subroutine is not provided to extract it.

The subroutines make no attempt to detect singularity in $\mathbf{X}$. This is usually indicated by large values for the parameter estimates, their variances and their correlations. The subroutines leave it to the user to make further analyses of the type described in Section 2.6 if a singularity is thought to be possible. Linear dependencies between the

columns of X do not arise in the specific applications of the least squares subroutines in this thesis. However in a general least squares system, another subroutine should be available to assess singularity; such a subroutine has not been implemented in this thesis.

3.MODELS WITH A SINGLE SYSTEMATIC PART

3.1 GENERALIZED LINEAR MODELS

This section discusses a class of models called Generalized Linear Models which includes the normal linear model and many other similar models involving distributions such as the binomial and Poisson distributions. The class of models has been widely used, largely because the computer programs GLIM and GENSTAT make it easy to specify and fit the models. This, in turn, has been possible because a standard iterative algorithm for maximum likelihood parameter estimation, called Fisher's scoring technique (FS), whose iterations are

$$\boldsymbol{\beta}^{+}_{FS} = \boldsymbol{\beta} - E\left[\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2}\right]^{-1} \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} ,$$

can be implemented with a similar series of weighted least squares calculations for all models in the class.

The method arose out of one for binomial data that was originated by Finney(1947). He proposed modelling responses which are numbers of successes out of r_i trials, using binomial distributions with parameters r_i and π_i where

$$\pi_i = \Phi(\mathbf{x}_i'\boldsymbol{\beta}) ,$$

$\Phi(\cdot)$ is the unit normal distribution function and $\mathbf{x}_i'\boldsymbol{\beta}$ is a linear function of explanatory variables and unknown parameters; this is called a probit model. He showed that the calculations for applying FS to the probit model could be expressed as a series of weighted least squares calculations. Nelder(1968) found a similar result for normal models involving inverse polynomials with $E[y_i] = (\mathbf{x}_i'\boldsymbol{\beta})^{-1}$.

These models and algorithms were generalized by Nelder and Wedderburn(1972) to a general exponential family of independent distributions which they called Generalized Linear Models (GLM's) and which have log probability density functions of the form

$$\log f(y_i \mid \boldsymbol{\beta}, \phi) = \alpha(\phi)\{y_i\theta_i - g(\theta_i)\} + h(\phi, y_i) \qquad (3.1.1)$$

where $\theta_i = k(\eta_i)$ and $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$. These models cover many standard distributions such as the binomial, Poisson, gamma and normal distributions and allow the mean $E[y_i] = g'(\theta_i)$ to be related to the linear part η_i in an arbitrary way. Nelder and Wedderburn(1972) describe several applications of the models; others are described by Manly and Crosbie(1977).

In this section we restrict attention to the special case where ϕ in (3.1.1) is a known value (such as in the binomial and Poisson distributions). In Chapter 4 we shall describe how the method can be extended to models which have extra unknown auxiliary parameters such as ϕ. When ϕ is a known constant, FS for $\boldsymbol{\beta}$ is equivalent to a procedure of iteratively reweighted least squares (IRLS) with weights w_i, "response" variable z_i^* and explanatory variables $\mathbf{x}_i$ where

$$w_i = g''(\theta_i)\{k'(\eta_i)\}^2$$

and

$$z_i^* = \eta_i + \frac{y_i - g'(\hat{\theta}_i)}{g''(\theta_i)\,k'(\eta_i)}$$

Since $g''(\theta_i) = \mathrm{var}(y_i)/\alpha(\phi)^2$, all weights are positive. The IRLS calculations do not involve ϕ and we shall see later that the same algorithm can also be used even when ϕ is unknown, though strictly this is not joint FS for $\boldsymbol{\beta}$ and ϕ.

The parameterization $\theta_i = \eta_i$ was called the "natural" parameterization by Nelder and Wedderburn(1972). With this parameterization, and fixed, known ϕ, the likelihood function is convex and does not have multiple local maxima; if the algorithm converges, it is to the unique maximum likelihood estimate of $\boldsymbol{\beta}$. It can also be shown that this FS algorithm converges quadratically once it is near

enough to the maximum likelihood estimate. For other relationships between θ_i and η_i (other link functions in GLIM terminology), the IRLS algorithm does not have these properties. In practice however, the algorithm usually seems to converge and problems with multiple local maxima seem rare.

Nelder and Wedderburn(1972) suggested solving $g'(\theta_i) = y_i$ (that is, $E[y_i] = y_i$) for θ_i and using one FS iteration with θ_i replaced by this in the formulae for w_i and z_i^*, in order to get starting values for $\boldsymbol{\beta}$.

Because of the simplicity of the algorithm and its ease of use in the GLIM system (and also in GENSTAT), much effort has been expended on expressing other models that are not GLMs in a form that allows the algorithm to be used. Two important examples of this type are described next.

Many multinomial models can be fitted by pretending that the cell counts have independent Poisson distributions. If the expected cell counts are $E[y_i] = \mu_i = \mu_i(\boldsymbol{\beta})$ for $i=1,\ldots,n$, then the kernel of the multinomial log-likelihood is

$$\ell(\boldsymbol{\beta}) = \sum y_i \log(\mu_i)$$

with the constraint, $\sum \mu_i = N$ where N is the known multinomial total. The maximum of this function with respect to $\boldsymbol{\beta}$ subject to $\sum \mu_i = N$, is identical to the maximum of

$$\ell^*(\boldsymbol{\beta}) = \sum y_i \log(\mu_i) - \sum \mu_i$$

with respect to $\boldsymbol{\beta}$ subject to $\sum \mu_i = N$. In some models $\sum \mu_i(\boldsymbol{\beta}) = N$ for all $\boldsymbol{\beta}$ and so the constraint is unnecessary. Most other models of importance (such as log-linear models) can be reparameterized in the form $\mu_i(\boldsymbol{\beta}) = \beta_1 \mu_i^*(\boldsymbol{\beta}_2)$ and it can be shown that the unconstrained maximum of $\ell^*(\boldsymbol{\beta})$ then satisfies $\sum \mu_i = N$. In both cases, maximum likelihood estimates can be found by the unconstrained maximization of $\ell^*(\boldsymbol{\beta})$ and this is the kernel of the log-likelihood of a sample of independent Poisson variables. If $\boldsymbol{\beta}$ is involved linearly in $\mu_i(\boldsymbol{\beta})$ the models can be treated as if they were Poisson GLMs (Nelder and

Wedderburn, 1972 and Nelder, 1974).

Aitkin and Clayton(1980) ingeniously showed that proportional hazards models for censored survival data with hazard function

$$h_i(y_i) = h_0(y_i).k(\mathbf{x}_i'\boldsymbol{\beta})$$

have a log-likelihood that can be written in the form

$$\ell(\boldsymbol{\beta}) = \sum \{w_i \log \mu_i - \mu_i\} + \sum \{w_i \log(h_0(y_i)/H_0(y_i))\}$$

where $H_0(y) = \int h_0(y)dy$, $\mu_i = H_0(y_i)k(\mathbf{x}_i'\boldsymbol{\beta})$ and where $w_i=0$ if the i'th observation is censored and $w_i=1$ otherwise. The last term does not involve $\boldsymbol{\beta}$ and can therefore be ignored; the kernel of $\ell(\boldsymbol{\beta})$ is then identical to the log-likelihood of the GLM where w_i is assumed to have a Poisson distribution, so that Nelder and Wedderburn's algorithm can be applied. (Auxiliary parameters in the base-line hazard function $h_0(y)$, however, do not usually factorize out in the way required for GLM's and alternative methods must be used to estimate them).

3.2 SOME MODELS OUTSIDE THE CLASS OF GENERALIZED LINEAR MODELS

Though the exponential family of Generalized Linear Models (GLM's) covers many standard types of models for independent responses, others outside that class are sometimes appropriate. A few examples are described in this section.

Some models involve distributions in an exponential family, but their parameters are not involved in the form required for GLM's.

(a) An extension to the normal linear model $y_i \sim N(\mathbf{x}_i'\boldsymbol{\beta})$ for $i=1,\ldots,n$, was proposed by Box and Cox(1964) for situations where the distribution of y_i is skew with its variance either increasing or decreasing with its mean. They suggested the model

$$y_i^{(\phi)} \sim N(\mathbf{x}_i'\boldsymbol{\beta}\ ,\ \sigma^2)$$

where

$$y^{(\phi)} = \begin{cases} (y^{\phi}-1)/\phi & \phi \neq 0 \\ \log(y) & \phi = 0 \end{cases}$$

(This is a reparameterization of an ordinary power transformation of the response to make it continuous in ϕ and y at $\phi=0$).

Manly(1976) proposed another model with similar properties but that can be used even if some $y_i<0$,

$$y_i^{[\phi]} \sim N(\mathbf{x}_i'\boldsymbol{\beta},\ \sigma^2)$$

where

$$y^{[\phi]} = \begin{cases} (\exp(\phi y)-1)/\phi & \phi \neq 0 \\ y & \phi = 0 \end{cases} .$$

Note that $y^{(\phi)} = (\log y)^{[\phi]}$. In both types of model, the unknown parameter ϕ does not fit into the GLM framework.

(b) Nonlinear normal models in which $E[y_i]$ is a nonlinear function of β are often used. Wedderburn(1974), Jennrich and Moore(1975) and Jorgensen(1983) discuss extensions of GLM's to nonlinear models.

(c) If $E[y_i] = x_i'\beta$ and y_i is normally distributed, the ordinary normal linear model can be modified to explain certain types of heteroscedasticity. If the variance of y_i increases (or decreases) with its mean $x_i'\beta$, then the model

$$y_i \sim N(x_i'\beta \, , \, \sigma^2 v(x_i'\beta))$$

can be used for some function $v(\cdot)$. Except on the rare occasion when $v(\cdot)$ can be fully specified prior to collecting the data, the function $v(\cdot)$ must incorporate auxiliary parameters, such as

$$v(x_i'\beta,\phi) = 1 + \phi.x_i'\beta$$
$$\text{or } v(x_i'\beta,\phi) = \exp\{\phi.x_i'\beta\}$$

In both cases, constant variance corresponds to $\phi=0$. Williams(1959), Finney and Phillips(1977) and Raab(1981) considered models of this form.

Another more general assumption suggested by Rutemiller and Bowers(1968) and Harvey(1976), that can also be used to explain non-constant variance is that $var(y_i)$ depends on explanatory variables in the vector x_i^* whose components are not necessarily the same as these in x_i. This model is

$$y_i \sim N(x_i'\beta \, , \, v(x_i^{*\prime}\beta^*)) \qquad (3.2.1)$$

where $v(\cdot)$ is commonly either the identity or exponential function. Linear models with random coefficients can also be expressed in this heteroscedastic form (Hildreth and Houck, 1968).

(d) Many responses that are counts have variances that are larger than expected under the Poisson distribution. The negative binomial distribution is often used as an alternative in these

circumstances. Applications in single-sample situations include accident proneness (Greenwood and Yule, 1920), consumer purchasing (Chatfield, 1969), medical consultations (Kilpatrick, 1977) and measuring wildlife populations (White and Eberhardt, 1980). With a random sample from a single negative binomial distribution, the parameterization used is unimportant. However, different parameterizations result in different regression models (Stirling, 1984). The models can be motivated in one of two ways :-

(1) The individual count y_i has a Poisson distribution with mean m_i, but m_i is also random, $m_i \sim (\mu_i/\phi_i)\text{Gamma}(\phi_i)$. The compound distribution of y_i, which Johnson and Kotz(1969) denote by

$$\text{Poisson}(m_i) \bigwedge_{m_i} \text{Gamma}(\phi_i \ , \ \mu_i/\phi_i) \ ,$$

is then negative binomial with mean μ_i and variance $\mu_i(1+\mu_i/\phi_i)$. The distribution is only in Nelder and Wedderburn's exponential family if all ϕ_i are the same and are known. In practice however, $\{\phi_i\}$ are rarely known and ϕ_i may also depend on μ_i.

(2) The measured count y_i is the sum of counts in r_i "clusters" where $r_i \sim \text{Poisson}(\lambda_i)$ and the number of individuals in each cluster has an independent log series distribution with parameter θ_i. The distribution of y_i, which Johnson and Kotz(1969) denote by

$$\text{Poisson}(\lambda_i) \bigvee \text{logseries}(\theta_i) \ ,$$

is negative binomial with mean μ_i and variance $\mu_i(1+\phi_i)$ where

$$\mu_i = -[\log(1-\theta_i)]^{-1}\lambda_i \quad \text{and} \quad \phi_i = \theta_i/(\theta_i-1)$$

The reasonable assumption of constant ϕ_i results in a distribution that is not a GLM; its variance is a different function of its mean from the relationship in (1). Other relationships between ϕ_i and μ_i are also possible.

In practice, the mechanism generating the data is rarely known well enough to allow the relationship between the mean and variance to be specified before the data are collected. If enough data are available, a more general negative binomial model could be tried which has, as special cases, the models with constant ϕ_i in (1) and (2) above and which is defined by the relationship

$$\text{var}(y_i) = \mu_i(\phi_1 + \phi_2\mu_i)$$

The two assumptions (1) and (2) together actually result in the distribution

$$\{\ \text{Poisson}(m_i) \bigwedge_{m_i} \text{Gamma}(\phi_i\ ,\ \lambda_i/\phi_i)\ \} \bigvee \text{logseries}(\theta_i)$$

which, though it is not exactly negative binomial, has the above relationship between its mean and variance (Johnson and Kotz, 1969, pages 125 and 204).

Another more general assumption that was used by Manton and Woodbury(1981) is found by specifying the shape parameter ϕ_i in (1) or (2) as a function of explanatory factors,

$$\phi_i = \phi(\mathbf{x}_i^{*\prime}\boldsymbol{\beta}^*)$$

where $\phi(\cdot)$ is some simple function such as the identity or exponential function and $\mathbf{x}_i^*$ may include explanatory variables that also affect the mean.

The various strategies for modelling the auxiliary parameter here (constant, depending on μ_i or depending on $\mathbf{x}_i^{*\prime}\boldsymbol{\beta}$) mirror the models discussed in (c) for the variance of the normal distribution.

(e) A problem that can arise with regression models for binomial data is that sometimes even extreme values of an explanatory variable result in a proportion ϕ of responses with $0 < \phi < 1$.

Finney(1944) described an extension of the probit model to include such "residual responses",

$$y_i \sim \text{binomial}\left(r_i\ ,\ \phi + (1-\phi)\ \Phi(\mathbf{x}_i'\boldsymbol{\beta})\ \right).$$

Similar modifications can be made to other types of model for binomial data, such as logit models.

Some commonly used models involve distributions that are not in an exponential family.

(f) If there are a few extreme errors (outliers), $(y_i-\mathbf{x}_i'\boldsymbol{\beta})$ that would be unlikely in the normal distribution, but the remaining standard assumptions of independence, linearity and constant variance all hold, then the assumption of normality can be replaced by another distribution with longer, thicker tails. A frequently used class of such thicker-tailed alternatives has log probability density functions of the form

$$\log f_i(y_i) = -\psi(|y_i-\mathbf{x}_i'\boldsymbol{\beta}|\sigma^{-1}) - \log\sigma - \log\{\int_{-\infty}^{\infty}\exp(-\psi(|z|))dz\ \}$$

(The resulting maximum likelihood estimators were called robust M-estimators by Huber(1964)). Standard distributions of this form are the double exponential distribution with

$$\psi(u) \propto u$$

and the t-distribution with ϕ degrees of freedom

$$\psi(u) = \frac{(\phi+1)}{2}\ .\ \log(1 + u^2/\phi)\ .$$

Another example is the distribution suggested by Huber with

$$\psi(u) = \begin{cases} u^2/2 & u<\phi \\ \phi|u| - \phi^2/2 & u\geq\phi\ . \end{cases}$$

Andrews et al(1972) compare several such estimators. Another distribution which is more robust than the distributions used in

that study, is the log-tailed distribution with

$$\psi(u) = \begin{cases} u^2/2 & u<\phi \\ \phi^2\log(u/\phi) + \phi^2/2 & u\geq\phi \end{cases}$$

where $\phi > 1$ (Stirling,1984). Most models of these types involve an extra unknown parameter ϕ. The only distribution in the GLM class that is symmetric for all values of the parameters is the normal distribution, so no robust models are GLM's.

(g) A similar problem to that discussed in (d) for Poisson counts may also arise with the binomial distribution; the actual variance is sometimes greater than the theoretical binomial variance. Skellam(1948) derived the compound distribution

$$\text{binomial}(r_i, p_i) \overset{p_i}{\wedge} \text{beta}(\alpha_i, \beta_i)$$

and showed that it was beta-binomial with probability function

$$p(y_i \mid \alpha_i, \beta_i) = \binom{r_i}{y_i} \frac{B(\alpha_i+y_i\ ,\ r_i+\beta_i-y_i)}{B(\alpha_i\ ,\ \beta_i)}$$

for $y_i = 0,1,\ldots,r_i$, where $B(\cdot,\cdot)$ is the beta function. This has mean $r_i\pi_i$ and variance $r_i\pi_i(1-\pi_i)\{1+(r_i-1)/(\alpha_i+\beta_i+1)\}$ where $\pi_i=\alpha_i/(\alpha_i+\beta_i)$. Examples of its use in a non-regression setting were given by Ishii and Hayakawa(1960), Chatfield and Goodhardt(1970) and Griffiths(1973). Crowder(1978) described a regression model of this form with

$$\pi_i = \exp(\mathbf{x}_i'\boldsymbol{\beta}) \,/\, (1 + \exp(\mathbf{x}_i'\boldsymbol{\beta}))$$

and the reasonable assumption of constant $(\alpha_i+\beta_i)$; other assumptions such as constant β_i are, however, also possible. None of the beta-binomial models are in Nelder and Wedderburn's exponential family.

(h) Another area to which much research has recently been directed but where GLMs are not appropriate is the analysis of survival or lifetime data. An important characteristic of this type of data

is that it is often censored so that the exact time of failure or death, y_i, is only known provided $y_i < t_i$; otherwise failure time is just recorded as "above t_i". Kalbfleisch and Prentice(1980, Chapter 2) describe several parametric distributions that have been used for survival data. These have been extended into regression models in two ways.

(1) Proportional hazards models were suggested by Cox(1972). If the i'th failure time has probability density function $f_i(y)$ and distribution function $F_i(y)$, then proportional hazards models assume that differences in explanatory variables have the same proportional effect on the hazard function $h_i(y) = f_i(y)/(1-F_i(y))$ for all y. The i'th individual's hazard function can therefore be written as

$$h_i(y_i) = h_0(y_i) \cdot k(\mathbf{x}_i'\boldsymbol{\beta})$$

for some functions $h_0(\cdot)$ and $k(\cdot)$.

(2) Accelerated failure time models assume that the explanatory variables act directly on the time scale so that

$$F_i(y_i) = F_0(y_i . k(\mathbf{x}_i'\boldsymbol{\beta}))$$

for some functions $F_0(\cdot)$ and $k(\cdot)$.

Models of these types have been used both with a parametric baseline distribution corresponding to $h_0(\cdot)$ or $F_0(\cdot)$ and also non-parametrically where the baseline distribution is unspecified. Usually $k(\cdot) = \exp(\cdot)$.

Some models involve correlated responses and therefore are not GLM's.

(i) Most models for correlated responses are based on the multivariate normal distribution,

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta} , \mathbf{V}(\boldsymbol{\phi}) \sigma^2)$$

The most common sources of this model are in experimental designs with random effects and in time series data. The most general random effects models can be written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \sum_{i=1}^{k} \mathbf{Z}_i \mathbf{e}_i$$

where $\mathbf{e}_i \sim N(\mathbf{0}, \mathbf{I}\sigma_i^2)$ for $i=1,\ldots,(k-1)$ are vectors of random effects, $\mathbf{Z}_k = \mathbf{I}$ and $\mathbf{e}_k$ is the vector of independent errors. Note that the vectors $\mathbf{e}_i$ are usually not all of the same length,

In time series data the observation y_i is assumed to be dependent in some way on previous values $y_{(i-1)}, y_{(i-2)}, \ldots$; although often correlated with $y_{(i+1)}, y_{(i+2)}, \ldots$, it is not causally affected by subsequent values in the time series. First order autoregressive (AR) models are such that

$$y_i = \mathbf{x}_i'\boldsymbol{\beta} + e_i \qquad \text{with } e_i = \phi e_{i-1} + e_i^*$$

where $\{e_i^*\}$ are independent $N(0,\sigma^2)$. First order moving average (MA) models are of the form

$$y_i = \mathbf{x}_i'\boldsymbol{\beta} + e_i + \phi e_{i-1}$$

where $\{e_i\}$ are independent $N(0,\sigma^2)$. The orders of AR and MA models can be increased and they can be combined to give ARMA models.

(j) The main non-normal distribution used in models for correlated responses is the multinomial distribution. This can arise when a nominal response or an ordinal response is to be related to explanatory variables (McCullagh(1980) and Anderson and Philips(1981)). It is also used in most models for contingency tables; depending on how the table was recorded, either the whole table is usually assumed to be multinomial or each of several layers is assumed to be independently multinomial (Bishop et al, 1975). In all cases, we can denote the data by y_{ij} for $i=1,\ldots,\ell$ and $j=1,\ldots,a$ where $[y_{i1},\ldots,y_{ia}]$ for $i=1,\ldots,\ell$ are independent multinomial vectors with means $[\mu_{i1},\ldots,\mu_{ia}]$ and

$\sum_j \mu_{ij} = n_i$.

Various models have been proposed to explain the dependence of μ_{ij} on explanatory variables. In contingency tables, log-linear models with

$$\mu_{ij} = \exp(\mathbf{x}_{ij}'\boldsymbol{\beta})$$

are usually appropriate. Bishop et al(1975), Fienberg(1980) and several other books on contingency tables describe many meaningful log-linear models. Occasionally linear models are appropriate, as in tables with marginal homogeneity (Wedderburn, 1974); the hypothesis of symmetry can be expressed either as a linear or log-linear model.

Multinomial models for ordinal responses are usually expressed in terms of an assumed underlying unobserved quantitative response z_i with some cumulative distribution function $\Phi(z_i - \mathbf{x}_i'\boldsymbol{\beta})$; it is assumed that ordinal response j is recorded if $\kappa_{j-1} < z_i \leq \kappa_j$ so that

$$\mu_{ij} = n_i \{ \Phi(\kappa_j - \mathbf{x}_i'\boldsymbol{\beta}) - \Phi(\kappa_{j-1} - \mathbf{x}_i'\boldsymbol{\beta}) \}.$$

In later sections of this thesis, algorithms will be developed that can be applied to many of the types of model described above.

3.3 GENERAL OPTIMIZATION ALGORITHMS FOR MAXIMUM LIKELIHOOD

We shall first express log-likelihoods in a particular way that will be useful when maximum likelihood algorithms are considered later. All multiparameter models that depend on a vector of p unknown parameters $\boldsymbol{\beta}$ have log-likelihoods that can be written in the form

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{m} \ell_i(\eta_i) + K \qquad (3.3.1)$$

where $\eta_i = \eta_i(\boldsymbol{\beta})$ for $i=1,\ldots,m$ are scalar functions called the systematic part of the model, K does not depend on $\boldsymbol{\beta}$ and $\ell_i(\cdot)$ for $i=1,\ldots,m$ are called log-likelihood components. Trivially we may always define $m=1$, $\ell_1(\eta_1) = \eta_1$, $K=0$ and $\eta_1(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta})$, but other definitions of the log-likelihood components with $m > 1$ are often more useful. The representation of any model in terms of a systematic part and log-likelihood components is not unique.

With independent discrete or continuous responses, the log-likelihood components would usually be taken to be the individual log probability functions or log probability density functions; mixed independent distributions such as censored survival distributions can be similarly expressed. For example in censored distributions, the exact failure time y_i is only known provided $y_i < t_i$; otherwise failure time is just recorded as "above t_i". If the probability density function of y_i is $f(y_i \mid \eta_i)$ then it corresponds to a log-likelihood component

$$\ell_i(\eta_i) = \begin{cases} \log f(y_i \mid \eta_i) & \text{if } y_i < t_i \\ \log \int_{t_i}^{\infty} f(y_i \mid \eta_i)\, dy_i & \text{if } y_i \geq t_i \,. \end{cases}$$

Many models for dependent observations also have log-likelihoods that can be written as a sum of simple log-likelihood components. For example, if $y_1,\ldots,y_n$ are counts with a multinomial or product

multinomial distribution and $E[y_i] = \eta_i(\boldsymbol{\beta})$ then we can define $\ell_i(\eta_i) = y_i \log \eta_i$. If a vector of n measurements, $\mathbf{y}$, is multivariate normal with mean depending on $\boldsymbol{\beta}$ and known variance-covariance matrix, then there exist independent linear functions of $\mathbf{y}$, $\mathbf{b}_i'\mathbf{y}$ for $i=1,\ldots,n$ and the log-likelihood components can be defined as the logarithms of the normal probability densities of these; if there are unknown parameters in the variance-covariance matrix then $\mathbf{b}_1,\ldots,\mathbf{b}_n$ in the log-likelihood components would be functions of these parameters.

Many models for time series data can be usefully written as a sum of log-likelihood components that are logs of the conditional probability (density) functions of y_i conditional on $y_1,\ldots,y_{(i-1)}$ for $i=1,\ldots,n$.

A final example here is the model suggested by Box and Cox(1964) in which transformations of n independent responses are assumed to satisfy an ordinary normal linear model, but where the transformations involve an unknown parameter. The log-likelihood can be written as the sum of (n+1) log-likelihood components where the first n are logarithms of normal probability density functions and the last is the Jacobian of the transformation of the responses.

The problem of maximizing a log-likelihood $\ell(\boldsymbol{\beta})$ is clearly a special case of general unconstrained optimization. General optimization algorithms can therefore be used to find maximum likelihood estimates. However the special structure of $\ell(\boldsymbol{\beta})$, which allows expectations to be taken, sometimes also allows other algorithms to be used. We next discuss general maximum likelihood algorithms.

Optimization algorithms can be broken into two classes, (a) algorithms that evaluate (or approximate) second derivatives of $\ell(\boldsymbol{\beta})$ and (b) those that do not. The latter methods include search methods such as the simplex method (Nelder and Mead, 1965) which are useful for discontinuous functions, and gradient methods such as steepest descent (Cauchy, 1847) and conjugate gradients (Hestenes and Stiefel, 1952). For large problems, they require much less storage than second

derivative methods and they may need to be used for this reason if many parameters must be estimated. However, they are often very slow to converge and therefore second derivative methods would be preferred in applications where these can be easily evaluated. Since the applications that will be examined later in this thesis are of this type, we now restrict attention to second derivative algorithms.

If $\boldsymbol{\beta}$ is close enough to the maximum likelihood likelihood estimate $\hat{\boldsymbol{\beta}}$ for a quadratic approximation to be used, then

$$\hat{\boldsymbol{\beta}} \approx \boldsymbol{\beta} - \mathbf{H}^{-1}\mathbf{g} \tag{3.3.2}$$

where $\mathbf{H} = \partial^2 \ell(\boldsymbol{\beta})/\partial\boldsymbol{\beta}^2$ and $\mathbf{g} = \partial\, \ell(\boldsymbol{\beta})/\partial\boldsymbol{\beta}$. All second derivative methods are based on iterations of the form

$$\boldsymbol{\beta}^{+} = \boldsymbol{\beta} - \mathbf{A}^{-1}\mathbf{g} \tag{3.3.3}$$

where $\mathbf{A}$ is a matrix closely related to $\mathbf{H}$.

Setting $\mathbf{A}=\mathbf{H}$ defines the Newton-Raphson (NR) algorithm. It converges quadratically when $\boldsymbol{\beta}$ is near $\hat{\boldsymbol{\beta}}$. In models with log-likelihood of the form (3.3.1), the iterations can be expressed as

$$\boldsymbol{\beta}^{+}_{NR} = \boldsymbol{\beta} - \left[\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial\, \boldsymbol{\beta}^2}\right]^{-1} \left[\frac{\partial\, \ell(\boldsymbol{\beta})}{\partial\, \boldsymbol{\beta}}\right] \tag{3.3.4}$$

$$= \boldsymbol{\beta} + (\mathbf{X}'\mathbf{W}\mathbf{X} - \mathbf{B})^{-1}\, \mathbf{X}'\mathbf{v} \tag{3.3.5}$$

where $\mathbf{X}$ has i'th row $\mathbf{x}_i' = \partial\eta_i/\partial\boldsymbol{\beta}'$, $\mathbf{v}$ is the vector with i'th element $v_i = \partial\ell_i(\eta_i)/\partial\eta_i$, $\mathbf{W}$ is a diagonal matrix with i'th diagonal element $w_i = -\partial^2\ell_i(\eta_i)/\partial\eta_i^2$ and $\mathbf{B} = \sum v_i(\partial^2\eta_i/\partial\boldsymbol{\beta}^2)$. The NR algorithm can have two problems leading to non-convergence, so that a modified method must be used in practice :-

Definiteness Modifications

If $\mathbf{H}$ is not negative definite, the full NR iterations will lead towards a minimum or saddlepoint of the likelihood. Three types of modification have been suggested, each of which entails subtracting a positive definite matrix $\mathbf{E}$ from $\mathbf{H}$ such that $\mathbf{A} = \mathbf{H}-\mathbf{E}$ is negative definite.

(i) $\mathbf{E} = \sum_{i=1}^{p} \max(\varepsilon - \lambda_i, 0)\mathbf{u}_i\mathbf{u}_i'$ where $\{\lambda_i\}$ and $\{\mathbf{u}_i\}$ are the eigenvalues and corresponding eigenvectors of $\mathbf{H}$, and ε is a small constant; this was suggested by Greenstadt(1967).

(ii) $\mathbf{E} = \max(\varepsilon - \lambda_1, \varepsilon - \lambda_2, \ldots, \varepsilon - \lambda_p, 0)\mathbf{I}$ was suggested by Goldfeld, et al(1966).

(iii) Gill and Murray(1974) described how to find a small diagonal matrix $\mathbf{E}$ which is sufficient to make $\mathbf{A}$ negative definite. $\mathbf{E}$ is found in the course of a Choleski factorization of $-\mathbf{H}$ and involves little extra work over that necessary to find $\mathbf{A}^{-1}\mathbf{g}$. The method ensures that all diagonal elements of the Choleski factorization are at least ε. Since (i) and (ii) involve computing eigenvalues and eigenvectors of $\mathbf{H}$, method (iii) is normally used.

One undesirable feature of all definiteness modifications above is that they are affected by the units in which the parameters are measured. In the introduction to the E04 optimization subroutines in the NAG subroutine library (1983) the following recommendation is therefore made.

> "Scaling (in a broadly defined sense) often has a significant influence on the performance of optimization methods. Since convergence tolerances and other criteria are necessarily based on an implicit definition of "small" and "large", problems with unusual or unbalanced scaling may cause difficulties for some algorithms. Nonetheless, there are currently no scaling routines in the library, although the position is under constant review. In light of the present state of the art, it is considered that sensible scaling by the user is likely to be more effective than any automatic routine.....
>
> .... One method of scaling is to transform the variables from their original representation, which may reflect the physical nature of the problem, to variables that have certain desirable properties in terms of optimization. It

is generally helpful for the following conditions to be satisfied:

(a) the variables are all of similar magnitude in the region of interest;

(b) a fixed change in any of the variables results in similar changes in (the function being maximized). Ideally, a unit change in any variable produces a unit change in (the function);

(c) the variables are transformed so as to avoid cancellation error in the evaluation of (the function). "

The requirement is not too critical in most examples though in extremely badly scaled examples, the above definiteness modifications may not result in an effective search direction or step size. The user must therefore give scaling some thought.

Stepsize modifications

Even when $\mathbf{A}$ is negative definite, $\boldsymbol{\beta}^{+}_{NR}$ might actually step too far and reduce the likelihood. Two modifications can be used.

(i) $\mathbf{A} = \alpha^{-1}(\mathbf{H} - \mathbf{E})$ where $0 \leq \alpha \leq 1$ and $\mathbf{E}$ is a definiteness modification if one is required. The simplest strategy is to halve α until $\boldsymbol{\beta}^{+}$ has higher likelihood than $\boldsymbol{\beta}$. Other search methods such as that of Bard(1974, pages 110-113) use quadratic approximations to update α.

(ii) $\mathbf{A} = (\mathbf{H} - \mathbf{E} - d\mathbf{I})$ where $d \geq 0$. This was suggested by Marquardt(1963) as a modification of the Gauss-Newton method for non-linear least squares but is equally applicable here; Marquardt's algorithm for updating d in each iteration can be used. The method has the advantage of approaching the steepest descent step direction as d is increased.

Definiteness and stepsize modifications are usually only required in the first few iterations since the quadratic approximation (3.3.2)

becomes more accurate as the solution is approached. The modified NR algorithm therefore becomes identical to unmodified NR once close enough to the maximum likelihood estimate. Although the modified NR method converges to a local maximum of the likelihood, this may not be a global maximum. The only satisfactory way to get some idea of whether a global maximum has been reached seems to be to perform the iterations from several starting values (unless theory can prove that there is a unique local maximum).

When $\mathbf{H}$ is difficult to evaluate, other definitions of $\mathbf{A}$ can be used. Fisher's scoring technique (FS) uses the approximation $\mathbf{A} = E[\mathbf{H}]$ so that $-\mathbf{A}$ is the Fisher's information matrix and

$$\boldsymbol{\beta}^{+}_{FS} = \boldsymbol{\beta} + \left(E[\mathbf{X}'\mathbf{W}\mathbf{X}|\mathbf{B}] \right)^{-1} \mathbf{X}'\mathbf{v} \qquad (3.3.6)$$

This definition of $\mathbf{A}$ does not tend to $\mathbf{H}$ as the maximum likelihood solution is approached and so has poorer convergence than NR near $\hat{\boldsymbol{\beta}}$; however it has the advantage that $\mathbf{A}$ is always negative semi-definite so that definiteness modifications are unnecessary to get convergence. Mantel and Myers (1971) found FS more likely to converge than unmodified NR in a problem with censored data, but that if both methods converged, NR was faster. These results may hold for other problems; further comparisons are made in later sections. Stepsize modifications are still needed for the FS algorithm to ensure convergence.

Another definition of $\mathbf{A}$ is given by the quasi-Newton (QN) method (Davidon, 1959). This uses the gradients and function evaluations in successive iterations to build up a negative definite approximation $\mathbf{A}$ to $\mathbf{H}$, so that $\mathbf{A} \rightarrow \mathbf{H}$ as the iterations progress. Like the FS algorithm, $\mathbf{A}$ is negative definite at each iteration and therefore definiteness modifications are unnecessary. However we shall be restricting attention in later sections of the thesis to algorithms that can be implemented with a sequence of least squares calculations. Since the QN algorithm does not fit into that framework, it is not examined further in this thesis.

A fourth type of definition of the matrix $\mathbf{A}$ in (3.3.3) is

described next. First $\eta_i^*(\cdot)$ is defined to be the first two terms of a Taylor series for $\eta_i(\cdot)$ round $\boldsymbol{\beta}$,

$$\eta_i^*(\boldsymbol{\beta}^+) = \eta_i(\boldsymbol{\beta}) + \mathbf{x}_i'(\boldsymbol{\beta}^+ - \boldsymbol{\beta})$$

where $\mathbf{x}_i$ is the i'th row of matrix $\mathbf{X}$ in (3.3.5). If $\ell^*(\boldsymbol{\beta})$ is used to denote the log-likelihood with $\eta_i(\cdot)$ replaced by $\eta_i^*(\cdot)$, then

$$\frac{\partial \ell^*(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \frac{\partial\, \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

and

$$\frac{\partial^2 \ell^*(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -\mathbf{X}'\mathbf{W}\mathbf{X} \quad .$$

If an iteration of NR to maximize $\ell^*(\boldsymbol{\beta})$ is used in each iteration rather than NR applied to $\ell(\boldsymbol{\beta})$, we call the algorithm a NRL algorithm, since it is found by applying NR to Linearized systematic parts, and

$$\boldsymbol{\beta}^+_{NRL} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\,\mathbf{X}'\mathbf{v} \qquad (3.3.7)$$

where $\mathbf{X}$, $\mathbf{W}$ and $\mathbf{v}$ are as in (3.3.5). This is an extension of the Gauss-Newton algorithm for nonlinear least squares and is of the form (3.3.3) with $\mathbf{A} = (\mathbf{X}'\mathbf{W}\mathbf{X})$. As mentioned at the start of this section, there is not a unique way to specify a model in terms of log-likelihood components and a systematic part. Whereas the NR and FS algorithms are not affected by this, redefining the systematic part leads to a different NRL algorithm, so that there is a whole class of NRL algorithms. Both definiteness and stepsize modifications may be needed.

We shall examine various properties of NRL algorithms, their implementation and their relationship to NR and FS in Sections 3.4 to 3.7. We end this section by describing one particular type of NRL algorithm that can be used for estimation in many models. The log-likelihood components are often bounded above by values a_i which may depend on $\mathbf{y}$, but not $\boldsymbol{\beta}$. If the systematic part $\eta_i(\boldsymbol{\beta})$ is defined to be the square root of (a_i minus the i'th log-likelihood component), then

$$\ell(\boldsymbol{\beta}) = -\sum \eta_i(\boldsymbol{\beta})^2 + K \qquad (3.3.8)$$

The NRL algorithm applied to this systematic part is then identical to the Gauss-Newton algorithm applied to this nonlinear least squares problem. This is the algorithm suggested by Ross(1982) and since $(\mathbf{X}'\mathbf{W}\mathbf{X})$ is positive semi-definite, definiteness modifications are not needed.

3.4 IMPLEMENTATION OF UNMODIFIED NRL ALGORITHMS

The main advantage of NRL algorithms over NR and FS is that their iterations can always be easily and accurately evaluated with weighted least squares calculations. Iterations of a NRL algorithm can be expressed in the form of weighted least squares calculations in two ways since

$$\boldsymbol{\beta}^{+}_{NRL} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\,\mathbf{X}'\mathbf{W}\mathbf{z} \qquad (3.4.1)$$

$$= (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\,\mathbf{X}'\mathbf{W}\mathbf{z}^{*} \qquad (3.4.2)$$

where $\mathbf{z} = \mathbf{W}^{-1}\mathbf{v}$, $\mathbf{z}^{*} = (\mathbf{X}\boldsymbol{\beta}+\mathbf{W}^{-1}\mathbf{v})$ and $\mathbf{X}$, $\mathbf{W}$ and $\mathbf{v}$ are as defined in (3.3.5). Equation (3.4.2) is in a similar form to Nelder and Wedderburn's iteratively reweighted least squares iterations for GLM's. However (3.4.1) is simpler and would be preferred in most applications; its weighted least squares calculations have "explanatory" variables $\mathbf{x}_i = \partial\eta_i(\boldsymbol{\beta})/\partial\boldsymbol{\beta}$, weights $w_i = -\partial^2\ell_i(\eta_i)/\partial\eta_i^2$ and "responses" $z_i = w_i^{-1}\partial\ell_i(\eta_i)/\partial\eta_i$ for $i=1,\ldots,n$.

When $\partial^2\ell_i(\eta_i)/\partial\eta_i^2 = 0$, which occurs if $\ell_i(\eta)$ is linear or has a point of inflection at η_i, the above formulae cannot be used. However the required effect for any such observation, which is to increase $\mathbf{X}'\mathbf{W}\mathbf{z}$ by $\{\partial\ell_i(\eta_i)/\partial\eta_i\}\mathbf{x}_i$ but to leave $\mathbf{X}'\mathbf{W}\mathbf{X}$ unaltered, can be found by replacing it with two pseudo-observations,

$$\begin{aligned} w_i^{(1)} &= 1 & w_i^{(2)} &= -1 \\ z_i^{(1)} &= \partial\ell_i(\eta_i)/\partial\eta_i & z_i^{(2)} &= 0 \\ \mathbf{x}_i^{(1)} &= \mathbf{x}_i & \mathbf{x}_i^{(2)} &= \mathbf{x}_i\ . \end{aligned} \qquad (3.4.3)$$

Although the basic NRL algorithm described above converges for most distributions and data sets, NRL does not guarantee convergence to a maximum of the likelihood. As discussed in Section 3.3, $(\mathbf{X}'\mathbf{W}\mathbf{X})$ should be positive definite at each iteration for the algorithm to converge. Even if $\mathbf{X}'\mathbf{W}\mathbf{X}$ is positive definite, an iteration may step too far and

reduce the likelihood. Definiteness and stepsize modifications may therefore be needed. In this section, we describe computational aspects of the basic unmodified algorithm. Implementation of definiteness and stepsize modifications is described in detail in Section 3.7.

If the weighted sums of squares matrix $\mathbf{X}'\mathbf{W}\mathbf{X}$ and vector $\mathbf{X}'\mathbf{W}\mathbf{z}$ (the Hessian and gradient) are formed, then the NRL iterations can be computed with standard definiteness and stepsize modifications using existing general optimization routines for solving (3.3.3), such as those in the NAG library. However it was shown in Chapter 2 that algorithms for least squares based on the sum of squares matrix can be very inaccurate when $\mathbf{X}$ is multicollinear. The main numerical advantage in representing iterations as weighted least squares calculations is that more accurate least squares algorithms based on QR factorizations can then be used.

For some definitions of the systematic parts, $\partial^2 \ell_i(\eta_i)/\partial \eta_i^2 < 0$ for all η_i and i; the systematic part η_i is then called convex. In models with convex systematic parts, all weights are positive in the least squares calculations, definiteness modifications are unnecessary and the NRL algorithm can be implemented using the least squares routines in many existing statistical computer packages. In models whose systematic part is not convex, some weights in the NRL algorithm may be negative and the standard QR algorithms must be modified to allow these. The necessary modifications to the Givens algorithm are described below; similar modifications to the other QR algorithms could be made.

Provided $[\mathbf{X} : \mathbf{z}]'\mathbf{W}[\mathbf{X} : \mathbf{z}]$ remains positive semidefinite as new rows of $[\mathbf{X} : \mathbf{z}]$ are incorporated, the Givens algorithm described in Section 2.3 can be used even with negative weights. It however loses numerical stability when a new row with negative weight changes $[\mathbf{X} : \mathbf{z}]'\mathbf{W}[\mathbf{X} : \mathbf{z}]$ from being positive definite to positive semidefinite or indefinite. (Some indefinite matrices do not even have Choleski factorizations). This problem may arise even when the final $[\mathbf{X} : \mathbf{z}]'\mathbf{W}[\mathbf{X} : \mathbf{z}]$ is positive definite, depending on the ordering of the rows of $[\mathbf{X} : \mathbf{z}]$.

The best solution is to separately accumulate the rows of $[\mathbf{X} : \mathbf{z}]$ with positive and negative weights, so that $\mathbf{R}_i$, $\mathbf{R}_2$, $\mathbf{D}_1$ and $\mathbf{D}_2$ are found where

$$[\mathbf{X} : \mathbf{z}]'\mathbf{W}[\mathbf{X} : \mathbf{z}] = \mathbf{R}_1'\mathbf{D}_1\mathbf{R}_1 - \mathbf{R}_2'\mathbf{D}_2\mathbf{R}_2 \ ,$$

$\mathbf{R}_1$ and $\mathbf{R}_2$ are upper triangular and $\mathbf{D}_1$ and $\mathbf{D}_2$ are diagonal with non-negative elements. If $[\mathbf{X} : \mathbf{z}]'\mathbf{W}[\mathbf{X} : \mathbf{z}]$ is not positive semi-definite, a definiteness modification may be necessary before the positive and negative parts can be combined; a method of doing this is described in Section 3.7. In situations where a definiteness modification is unnecessary, we can find an upper triangular matrix $\mathbf{R}$ and a diagonal matrix $\mathbf{D}$ such that

$$\mathbf{R}'\mathbf{D}\mathbf{R} = \mathbf{R}_1'\mathbf{D}_1\mathbf{R}_1 - \mathbf{R}_2'\mathbf{D}_2\mathbf{R}_2$$

It is a fairly easy matter to calculate $\mathbf{R}$ and $\mathbf{D}$ since $\mathbf{R}'\mathbf{D}\mathbf{R} = \mathbf{R}^{*}{}'\mathbf{D}^{*}\mathbf{R}^{*}$ where

$$\mathbf{R}^{*} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{D}^{*} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & -\mathbf{D}_2 \end{bmatrix} .$$

Givens rotations can therefore be applied to the rows of $\mathbf{R}^{*}$ (with weights given by $\mathbf{D}^{*}$) to reduce it to upper triangular. Subroutine COMBINE, which is listed in Appendix B, determines $\mathbf{R}$ and $\mathbf{D}$ from $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{D}_1$ and $\mathbf{D}_2$. It returns an error code (IFAULT=2) if a definiteness modification is needed.

Once $\mathbf{R}$ has been evaluated, the NRL step for the p unknown parameters $\boldsymbol{\beta}$ can be found from it in exactly the same way as in ordinary least squares. In particular, if the first p rows of $\mathbf{R}$ are $[\mathbf{R}_1 : \mathbf{r}_2]$ with $\mathbf{R}_1$ unit upper triangular, then the NRL step is given by

$$\mathbf{R}_1 (\boldsymbol{\beta}^{+}_{\mathrm{NRL}} - \boldsymbol{\beta}) = \mathbf{r}_2 \qquad (3.4.4)$$

which is equivalent to (2.3.2) and can also be easily solved by back-substitution using subroutine BSUB in Appendix B.

There is no general way to obtain starting values for the NRL algorithm that can be used for all models, and in some types of models,

the modeller must supply an initial guess at the values of some or all parameters. This can however be avoided in models where the parameters occur linearly, so that $\eta_i = \eta_i(\nu_i)$ where $\nu_i = \mathbf{x}_i'\boldsymbol{\beta}$ and $\mathbf{x}_i$ may depend on $\mathbf{y}$ but not on $\boldsymbol{\beta}$ for $i=1,\ldots,n$. In these models, the NRL least squares calculation (3.4.2) only depends on $\boldsymbol{\beta}$ through the vector $\boldsymbol{\nu}$ with components $\nu_1,\ldots,\nu_n$. Stirling (1984) suggested using formula (3.4.2) in a preliminary iteration with $\boldsymbol{\nu}$ replaced by an estimate from a related model that can be more easily fitted, in order to get a starting value for $\boldsymbol{\beta}$. In particular, a saturated model in which each ν_i is treated as a separate parameter, can sometimes be used; simple explicit maximum likelihood estimates for $\boldsymbol{\nu}$ can often be found for saturated models and this is closely related to the method used to get starting values for generalized linear models by Nelder and Wedderburn (1972). If a sequence of models is to be fitted, it is usually an improvement to use the value of $\boldsymbol{\nu}$ from the last iteration of the most similar model already fitted. Other examples of the use of simpler related models to get starting values, are described for particular examples in later sections.

In models whose parameters are not involved linearly, (3.4.2) depends on the value of $\boldsymbol{\beta}$ from the previous iteration both through $\boldsymbol{\eta}(\boldsymbol{\beta})$ and also $\mathbf{X}(\boldsymbol{\beta})$, so the above technique for obtaining starting values cannot be used; a starting value for $\boldsymbol{\beta}$ must be obtained in some other way.

If $\boldsymbol{\beta}$ can be partitioned, $\boldsymbol{\beta}' = [\boldsymbol{\beta}_1' : \boldsymbol{\beta}_2']$ and the systematic part η_i can be written as a function of $\boldsymbol{\beta}_1$ and $\nu_i = \mathbf{x}_i'\boldsymbol{\beta}_2$ only, then the least squares calculation in (3.4.2) also only depends on $\boldsymbol{\beta}_1$ and $\nu_1,\ldots,\nu_n$. If the user can give a preliminary value to $\boldsymbol{\beta}_1$ with either a guess, any (possibly inefficient) estimation method or with a fixed "moderate" value, then the method above could be used, with $\boldsymbol{\beta}_1$ assumed to be known and fixed, to get starting values for $\nu_1,\ldots,\nu_n$. These, with the preliminary value of $\boldsymbol{\beta}_1$, can be used in (3.4.2) to get a starting value for all parameters.

If $\boldsymbol{\beta}$ can be partitioned as above and the systematic part depends

only on β_1 and $\nu_i = x_i(\beta_1)'\beta_2$ then, if β_1 is assumed to be known and fixed, the parameters β_2 occur linearly and a starting value for β_2 can be found from (3.4.2) applied to this constrained model. The user therefore again only needs to specify a starting value for β_1. This technique of obtaining starting vaules is related to methods that will be discussed in Chapter 5 and in particular Section 5.3.

We end this section by describing the computations involved for a Poisson example. For some Poisson data, the number of events recorded (such as accidents) is the total of events of different types caused by different factors. The Poisson mean is then the sum of the unknown component means and attempting to use explanatory variables to describe these leads to a linear model with $E[y_i] = x_i'\beta$. We consider two definitions of the systematic parts,

$$\text{(A1)}\quad \eta_i = x_i'\beta \qquad \ell_i(\eta_i) = y_i\log(\eta_i) - \eta_i$$

and

$$\text{(A2)}\quad \eta_i = \log(x_i'\beta) \qquad \ell_i(\eta_i) = y_i\eta_i - \exp(\eta_i) \quad .$$

Direct application of formula (3.4.1) to implement the NRL algorithms with these two systematic parts involves sequences of weighted least squares calculations with "explanatory" variables,

$$\text{(A1)}\quad x_i \qquad\qquad \text{(A2)}\quad (x_i'\beta)^{-1}x_i$$

weights,

$$\text{(A1)}\quad w_i = y_i(x_i'\beta)^{-2} \qquad\qquad \text{(A2)}\quad w_i = x_i'\beta$$

and "resonses",

$$\text{(A1)}\quad z_i = x_i'\beta(1 - (x_i'\beta)/y_i) \qquad \text{(A2)}\quad z_i = (y_i/(x_i'\beta) - 1)$$

The calculations for (A2) are simplified and comparisons with (A1) become easier if its i'th explanatory and response variables are multiplied by $(x_i'\beta)$ and its i'th weight is divided by $(x_i'\beta)^2$. Both NRL algorithms then use "explanatory" variables x_i and the "responses" for both algorithms are given by the formula

$$z_i = (y_i/(x_i'\beta) - 1) \;/\; w_i$$

where the weights are

$$\text{(A1)}\quad w_i = y_i(\mathbf{x}_i'\boldsymbol{\beta})^{-2} \qquad \text{(A2)}\quad w_i = (\mathbf{x}_i'\boldsymbol{\beta})^{-1}$$

For the algorithm (A1) based on systematic part $(\mathbf{x}_i'\boldsymbol{\beta})$, $w_i = 0$ if $y_i = 0$ and any such observation must be replaced by two pseudo-observations as described by equation (3.4.3). Starting values can be found from similar least squares calculations with responses z_i replaced by $z_i^* = z_i+(\mathbf{x}_i'\boldsymbol{\beta})$ and $(\mathbf{x}_i'\boldsymbol{\beta})$ replaced by $\max(y_i,0.5)$, the estimate from the saturated model, throughout. Systematic part (A2) is convex and so is systematic part (A1) if all $y_i>0$. When all $y_i>0$, both algorithms can therefore be implemented using the standard weighted least squares algorithms in existing statistical computer packages.

TABLE 3.4.1
Artificial Data for Fitting Poisson Model

Data Set 1		Data Set 2	
Count, y_i	Explanatory Variable, d_i	Count, y_i	Explanatory Variable, d_i
1	1	1	1
10	2	10	2
7	3	7	3
4	4	4	4
1	5	0	5

Both NRL algorithms have reasonably good convergence for many Poisson data sets, but algorithm (A2) can be poor at times. For example, both algorithms were applied to fit the model with $\mathbf{x}_i'\boldsymbol{\beta} = \beta_1+\beta_2 d_i$ to Data Set 1 in Table 3.4.1 and their iterations are shown in Table 3.4.2. Algorithm (A1) converges considerably better than (A2). A more extreme example is given by data set 2 in Table

3.4.1. The iterations of the two algorithms applied to this example are shown in Table 3.4.3. Algorithm (A2) has intolerably bad convergence here and was still not satisfactorily close to the maximum likelihood estimate after 50 iterations; algorithm (A1) would clearly be preferred.

TABLE 3.4.2

Iterations of two NRL Algorithms Applied to Data Set 1 in Table 3.4.1 Starting Values (Iteration 0) were Found as Decribed in the Text.

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C †
Algorithm (A1) - NR				
0	2.11611	-0.03607	-13.09459	
1	3.96726	-0.29548	-8.24195	0.70054
2	6.31968	-0.76167	-6.54386	0.42911
3	7.78554	-1.09096	-6.30533	0.15269
4	8.04400	-1.14869	-6.30008	0.02259
5	8.05000	-1.15000	-6.30008	0.00054
6	8.05000	-1.15000	-6.30008	0.00000
Algorithm (A2) - FS				
0	2.11611	-0.03607	-13.09459	
1	6.52800	-0.64267	-6.46860	0.28794
2	7.34319	-0.91440	-6.33858	0.46439
3	7.72145	-1.04048	-6.30867	0.46483
4	7.89676	-1.09892	-6.30198	0.46643
5	7.97836	-1.12612	-6.30050	0.46749
6	8.01647	-1.13882	-6.30017	0.46806
7	8.03430	-1.14477	-6.30010	0.46834
8	8.04264	-1.14755	-6.30008	0.46848
9	8.04655	-1.14885	-6.30008	0.46854
10	8.04838	-1.14946	-6.30008	0.46857
11	8.04924	-1.14975	-6.30008	0.46859
12	8.04965	-1.14988	-6.30008	0.46859
13	8.04983	-1.14994	-6.30008	0.46860

† the convergence rate is defined by (3.5.1)

Different NRL algorithms can therefore have very different convergence properties. In Sections 3.5 and 3.7, we therefore examine aspects of the convergence of the various possible NRL algorithms.

TABLE 3.4.3

Iterations of two NRL Algorithms Applied to Data Set 2 in Table 3.4.1 Starting Values (Iteration 0) were Found as Decribed in the Text.

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C †
Algorithm (A1) - NR				
0	1.20000	0.65000	-12.79249	
1	5.01491	-0.56358	-9.27143	0.59352
2	8.70058	-1.59495	-7.82066	0.35756
3	10.48431	-2.05227	-7.62329	0.14637
4	10.78402	-2.12854	-7.61927	0.02158
5	10.79063	-2.13021	-7.61926	0.00047
6	10.79064	-2.13021	-7.61926	0.00000
Algorithm (A2) - FS				
0	2.60831	-0.40467	-18.21671	
1	8.94279	-1.51426	-7.80305	0.24867
2	9.37642	-1.65881	-7.72896	0.76533
3	9.66128	-1.75376	-7.69016	0.79858
4	9.86010	-1.82003	-7.66787	0.82395
5	10.00562	-1.86854	-7.65412	0.84362
6	10.11619	-1.90540	-7.64514	0.85915
7	10.20274	-1.93425	-7.63902	0.87167
8	10.27216	-1.95739	-7.63469	0.88192
9	10.32896	-1.97632	-7.63153	0.89045
10	10.37622	-1.99207	-7.62917	0.89764
11	10.41610	-2.00537	-7.62738	0.90377
12	10.45016	-2.01672	-7.62598	0.90906
13	10.47956	-2.02652	-7.62488	0.91366
14	10.50516	-2.03505	-7.62400	0.91769
15	10.52765	-2.04255	-7.62329	0.92124
16	10.54753	-2.04918	-7.62271	0.92441
17	10.56522	-2.05507	-7.62223	0.92723
18	10.58105	-2.06035	-7.62183	0.92977
19	10.59529	-2.06510	-7.62149	0.93206
20	10.60816	-2.06939	-7.62121	0.93413
:	:	:	:	:
:	:	:	:	:

† the convergence rate is defined by (3.5.1)

3.5 ASYMPTOTIC PERFORMANCE OF NRL ALGORITHMS

The NR and NRL algorithms have iterations which were shown in (3.3.5) and (3.3.7) to have the form

$$\boldsymbol{\beta}^{+}_{NR} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{W}\mathbf{X}-\mathbf{B})^{-1}\mathbf{X}'\mathbf{v}$$
$$\boldsymbol{\beta}^{+}_{NRL} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{v}$$

where $\mathbf{X}$ has i'th row $\partial\eta_i/\partial\boldsymbol{\beta}'$, $\mathbf{W}$ is diagonal with i'th diagonal element $w_i = -\partial^2\ell_i/\partial\eta_i^2$, $\mathbf{v}$ is a vector with i'th element $v_i = \partial\ell_i/\partial\eta_i$ and

$$\mathbf{B} = \sum v_i \frac{\partial^2\eta_i}{\partial\boldsymbol{\beta}^2}$$

The NR algorithm has quadratic convergence since it reaches the maximum likelihood estimate in one step if the log-likelihood is quadratic and since the log-likelihood is approximately quadratic in a small enough neighbourhood of a local maximum of the likelihood. If we define $\delta(\boldsymbol{\beta}) = \sum |\beta_i - \hat{\beta}_i|$ where $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimate, then the rate of convergence can be measured by

$$C = \delta(\boldsymbol{\beta}^{+}) \;/\; \delta(\boldsymbol{\beta}) \quad . \tag{3.5.1}$$

The quadratic convergence of NR implies that C_{NR} tends to zero as the iterations progress. All algorithms of the form $\boldsymbol{\beta}^{+} = \boldsymbol{\beta}+\mathbf{A}^{-1}\mathbf{X}'\mathbf{v}$ in which $\mathbf{A}$ does not tend to the second derivative matrix as the iterations progress, have linear convergence and therefore C tends to some non-zero constant. Asymptotic convergence can be very slow depending on the difference between $\mathbf{A}$ and the second derivative matrix. The asymptotic performance of NRL algorithms therefore depends on the size of $\mathbf{B}$ at the maximum likelihood estimate. An asymptotic convergence rate above 0.5 is generally unacceptable, whereas an asymptotic rate less than 0.1 gives approximately one extra significant decimal digit in every iteration and is fast enough for most purposes. Poor asymptotic convergence rates not only result in excessive numbers of iterations, but also make it difficult to determine the numerical accuracy of computed solutions when the iterations are stopped and, as

will be shown in Section 3.9, can result in poor estimates of the variances of the parameter estimates.

The systematic part of a model is called <u>linear</u> when $\eta_i(\boldsymbol{\beta}) = \mathbf{x}_i'\boldsymbol{\beta}$ for i=1,...,n where the vectors $\mathbf{x}_1,\ldots,\mathbf{x}_n$ may depend on $\mathbf{y}$ but not on $\boldsymbol{\beta}$. Many common multiparameter models have linear systematic parts and Nelder and Wedderburn (1972) and Stirling(1984) give several examples. For linear models $\mathbf{B} = \mathbf{0}$ and so NR = NRL. In these models,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1' \\ \vdots \\ \mathbf{x}_n' \end{bmatrix}$$

which does not change from iteration to iteration and this is another simplification that makes NRL (and NR) attractive.

The other important special case which results in $\mathbf{B} = \mathbf{0}$ at the maximum likelihood estimate, is if $v_i = 0$ for i=1,...,n. This only occurs for sets of data where the model is a "perfect" fit to the data, in the sense that $\eta_i(\hat{\boldsymbol{\beta}})$ is equal to $\hat{\eta}_i$ for all i where $\hat{\eta}_1,\ldots,\hat{\eta}_n$ are the maximum likelihood estimates from the saturated model in which each η_i is treated as a separate parameter. This condition is a generalization of the standard result from nonlinear least squares that the Gauss-Newton algorithm (which is a NRL algorithm) has quadratic convergence if all residuals are zero (for example see Chambers, 1973). Though NRL is often not identical to NR in models that fit perfectly, its iterations approach those of NR as the maximum likelihood estimate is approached.

In models that do not fit perfectly and that have a nonlinear systematic part, NRL usually does not have quadratic convergence. However $E[\mathbf{B}] = \mathbf{0}$ and therefore as the sample size n increases, subject to certain regularity conditions, $\mathbf{B}$ becomes negligible when the data do come from the model being fitted. Therefore if an appropriate model is being fitted to a large enough set of data, NRL converges almost as fast as NR.

With smaller sets of data or when fitting inappropriate models, **B** can be large enough to seriously slow down convergence. Generally the size of **B** is mostly determined by the amount of nonlinearity in the systematic part. Often the size of **B** can be reduced, and convergence speeded, by choice of a systematic part that is as close as possible to linearity. The effect of **B** on the difference between NR and NRL also depends on the conditioning of the second derivative matrix. If the second derivative matrix is nearly singular, even subtracting a small **B** can cause a large change in its inverse, leading to big differences in step size and direction of NRL from NR.

We next illustrate the effects of these factors with some models in which y_i is assumed to be binomial with parameters r_i (the number of trials) and π_i (the probability of success in a single trial) for i=1,...,n. In the most common binomial models,

$$\pi_i = \Phi(\mathbf{x}_i'\boldsymbol{\beta}) \qquad (3.5.2)$$

where $\Phi(\cdot)$ is the cumulative distribution function of some standard distribution (usually the logistic or normal distribution). We shall initially consider probit models for which

$$\Phi(y) = \int_{-\infty}^{y} (2\pi)^{-1/2} \exp(-y^2/2)\, dy \quad .$$

The three NRL algorithms that will be examined are defined from systematic parts,

$$\text{(A3)} \quad \eta_i = \mathbf{x}_i'\boldsymbol{\beta}$$

$$\text{(A4)} \quad \eta_i = \log\left(\Phi(\mathbf{x}_i'\boldsymbol{\beta}) \,/\, (1.0 - \Phi(\mathbf{x}_i'\boldsymbol{\beta})) \right)$$

and $$\text{(A5)} \quad \eta_i = \pi_i = \Phi(\mathbf{x}_i'\boldsymbol{\beta}) \quad .$$

Systematic part (A3) is linear and (A4) is nearly linear in the central range of values of $\mathbf{x}_i'\boldsymbol{\beta}$. (It would have been exactly linear if $\Phi(\cdot)$ had been the cumulative distribution function of the logistic distribution and the logistic and normal distributions have similar shape.) Systematic part (A5) is the most highly nonlinear.

Direct application of (3.4.1) to these three systematic parts leads to least squares calculations with "explanatory" variables (A3) $\mathbf{x}_i$,

(A4) $\phi(\mathbf{x}_i'\boldsymbol{\beta})/\{\pi_i(1-\pi_i)\}\mathbf{x}_i$ and (A5) $\phi(\mathbf{x}_i'\boldsymbol{\beta})\mathbf{x}_i$ where $\phi(\cdot)$ is the unit normal probability density function. In order to make comparisons easier and to avoid recalculating the explanatory variables in each iteration, the "explanatory" and "response" variables can be rescaled with corresponding changes to the weights, so that each NRL least squares calculation has "explanatory" variables $\mathbf{x}_i$. The weights for the three NRL algorithms are then

$$(A3)\quad w_i = \left\{ \frac{y_i}{\pi_i^2} + \frac{(r_i-y_i)}{(1-\pi_i)^2} \right\} \phi(\mathbf{x}_i'\boldsymbol{\beta})^2 - \left\{ \frac{y_i}{\pi_i} - \frac{(r_i-y_i)}{(1-\pi_i)} \right\} \phi'(\mathbf{x}_i'\boldsymbol{\beta})$$

$$(A4)\quad w_i = \frac{r_i\ \phi(\mathbf{x}_i'\boldsymbol{\beta})^2}{\pi_i\ (1-\pi_i)}$$

$$(A5)\quad w_i = \left\{ \frac{y_i}{\pi_i^2} + \frac{r_i-y_i}{(1-\pi_i)^2} \right\} \phi(\mathbf{x}_i'\boldsymbol{\beta})^2$$

and the "responses" in each case are

$$z_i = \left\{ \frac{y_i}{\pi_i} - \frac{(r_i-y_i)}{(1-\pi_i)} \right\} \phi(\mathbf{x}_i'\boldsymbol{\beta}) \ / \ w_i$$

Systematic parts (A4) and (A5) are convex and the corresponding NRL algorithms can be implemented in many existing statistical computer packages. For all NRL algorithms, a starting value can be obtained from (3.4.2) with π_i replaced by its estimate from the saturated model, y_i/r_i and $\mathbf{x}_i'\boldsymbol{\beta}$ similarly replaced by $\phi^{-1}(y_i/r_i)$. The starting value for $\boldsymbol{\beta}$ is then the regression coefficient from a least squares calculation which is of the same form as the ordinary NRL iterations except that the "response" z_i is replaced by $z_i^* = y_i/r_i + z_i$. In these calculations to get starting vaules, $y_i=0$ must be replaced by $y_i=1/2$ and $y_i=r_i$ must be replaced by $y_i=r_i-1/2$ to avoid infinite weights.

None of the numerical examples that follow need definiteness or stepsize modifications to get convergence, so the illustrations below are all based on unmodified NRL algorithms. The three algorithms were first applied to fit the probit model with

$$\mathbf{x}_i'\boldsymbol{\beta} = \beta_1 + \beta_2 d_i \qquad (3.5.3)$$

to the data in Table 3.5.1 and their iterations are shown in Table 3.5.2. Since the systematic part (A3) is linear, the corresponding NRL

TABLE 3.5.1

Artificial Data for Fitting Probit Model

Number of Trials, r_i	Number of Successes, y_i	Explanatory Variable, d_i
5	1	-2
5	0	-1
5	1	0
5	3	1
5	3	2
5	4	3

TABLE 3.5.2

Iterations of Three NRL Algorithms Applied to Data in Table 3.5.1. Starting Values (Iteration 0) were Found as Decribed in the Text.

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C
Algorithm (A3) - NR				
0	-0.51378	0.42602	-1.60402	0.04439
1	-0.54181	0.44108	-1.59722	0.01291
2	-0.54217	0.44128	-1.59722	0.00000
Algorithm (A4) - FS				
0	-0.52224	0.42964	-1.60085	0.03210
1	-0.54245	0.44164	-1.59722	0.02020
2	-0.54215	0.44127	-1.59722	0.05442
3	-0.54217	0.44128	-1.59722	0.04755
Algorithm (A5)				
0	-0.46521	0.40522	-1.64266	0.11492
1	-0.53807	0.43626	-1.59771	0.08074
2	-0.54098	0.44045	-1.59724	0.22223
3	-0.54195	0.44111	-1.59722	0.19041
4	-0.54213	0.44125	-1.59722	0.19802
5	-0.54216	0.44128	-1.59722	0.19628

algorithm is also NR and its convergence rate, C, tends to zero. (The estimate in the second NR iteration was not improved on by subsequent iterations of the algorithm.) The convergence rates of the other two NRL algorithms tend to non-zero values. Since systematic part (A4) is nearly linear, its asymptotic convergence rate, 0.05, is also good and its performance is similar to that of (A3). Systematic part (A5) is more highly nonlinear and its convergence rate, 0.20, is therefore relatively poor.

TABLE 3.5.3

Iterations of Three NRL Algorithms Applied to a Probit Model that Fits Perfectly from Starting Values $\beta_1 = \Phi^{-1}(0.4)$ and $\beta_2 = 0$

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C
Algorithm (A3) - NR				
0	-0.25335	0.00000		
1	-0.02796	0.83341	-0.33674	0.15522
2	-0.00106	0.99215	-0.00071	0.04578
3	0.00000	0.99998	0.00000	0.00234
4	0.00000	1.00000	0.00000	0.00000
Algorithm (A4) - FS				
0	-0.25335	0.00000		
1	0.00549	0.86154	-0.21896	0.11486
2	0.00031	0.99071	-0.00096	0.06664
3	0.00000	0.99995	0.00000	0.00503
4	0.00000	1.00000	0.00000	0.00002
Algorithm (A5)				
0	-0.25335	0.00000		
1	-0.13135	0.76403	-1.07480	0.29307
2	0.01175	1.03913	-0.02003	0.13852
3	0.00033	1.00134	-0.00002	0.03290
4	0.00000	1.00000	0.00000	0.00109
5	0.00000	1.00000	0.00000	0.00007

To illustrate the quadratic convergence of all NRL algorithms to data that a model fits perfectly, the probit model (3.5.3) was fitted to nine observations with $r_i = 10$, $y_i = i$ and $d_i = \Phi^{-1}(y_i/r_i)$ for $i=1,\ldots,9$. Unfortunately, if the starting values are chosen as described above from the saturated model, all NRL algorithms converge

in one iteration, so the starting values $\hat{\pi}_i = 0.4$ for all i were used instead to illustrate the algorithms' convergence rates. Table 3.5.3 shows the iterations of the three NRL algorithms. Though there are some differences between the algorithms, all display quadratic convergence, with C approaching zero.

We shall next illustrate the near-quadratic convergence of all NRL algorithms with large amounts of data. If there are repeated binomial observations at any value of $\mathbf{x}_i$, say with y_{ij} successes from r_{ij} trials for $j=1,\ldots,n_i$, then since w_i and $w_i z_i$ are both linear in r_{ij} and y_{ij} for all NRL algorithms, the iterations are identical to those obtained from agregating the repeated observations into similar binomial observations with $y_{i+} = \sum_j y_{ij}$ successes out ot $r_{i+} = \sum_j r_{ij}$ trials. For the illustration, we therefore use the probit model (3.5.3) with $d_i = -5, -4, \ldots, 4, 5$, and examine the effect of increasing the number of trials, r_i at each d_i. Threre sets of binomial observations were generated from the model with $\beta_1 = 0$ and $\beta_2 = 0.4$ with r_i set equal to 10, 100 and 1000 in turn. The starting values of $\hat{\pi}_i = 0.4$ were used in all cases rather than those from the saturated model, to allow a better comparison of the convergence rates. Tables 3.5.4 to 3.5.6 show the iterations of the three NRL algorithms. As expected, the asymptotic convergence rate of algorithm (A3) is zero for all r_i, and the asymptotic convergence rates of algorithms (A4) and (A5) approach zero as r_i increases.

As a final example in this section, we consider the residual responses model in which (3.5.2) is replaced by

$$\pi_i = \beta_1 + (1-\beta_1)\,\Phi(\mathbf{x}_i'\boldsymbol{\beta}_2) \quad , \qquad (3.5.4)$$

where $\Phi(\cdot)$ can be any comulative distribution function, such as those of the unit normal or logistic distributions. This model cannot be written with a single linear systematic part and so NR is not equivalent to any of the NRL algorithms described in this chapter. We consider here the two NRL algorithms defined from the systematic parts

TABLE 3.5.4

Iterations of Three NRL Algorithms Fitting a Probit Model to 11 Randomly Generated Binomial Values with r_i=10.

Starting Values $\beta_1 = \Phi^{-1}(0.4)$ and $\beta_2 = 0$.

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C
Algorithm (A3) - NR				
0	-0.25335	0.00000		
1	-0.04214	0.24612	-6.23701	0.32288
2	0.03367	0.34259	-3.48840	0.21003
3	0.05324	0.36646	-3.38172	0.05155
4	0.05430	0.36775	-3.38144	0.00272
5	0.05430	0.36776	-3.38144	0.00001
Algorithm (A4) - FS				
0	-0.25335	0.00000		
1	0.00314	0.26119	-5.34837	0.23353
2	0.03837	0.34089	-3.49633	0.27135
3	0.05328	0.36527	-3.38235	0.08185
4	0.05434	0.36769	-3.38144	0.02843
5	0.05430	0.36775	-3.38144	0.02292
6	0.05430	0.36776	-3.38144	0.01976
Algorithm (A5)				
0	-0.25335	0.00000		
1	-0.16952	0.20914	-9.79234	0.56622
2	0.09130	0.45035	-4.29472	0.31274
3	0.07481	0.38007	-3.41399	0.27442
4	0.05614	0.36657	-3.38169	0.09215
5	0.05421	0.36798	-3.38145	0.10436
6	0.05432	0.36772	-3.38144	0.17618
7	0.05430	0.36776	-3.38144	0.17792
8	0.05430	0.36775	-3.38144	0.17802

(A6) $\log(\pi_i/(1-\pi_i))$

(A7) π_i

Both can be expressed as IRLS with "explanatory" variables $\mathbf{X}$ whose i'th row is

$$\frac{\partial \pi_i}{\partial \begin{bmatrix} \beta_1 \\ \boldsymbol{\beta}_2 \end{bmatrix}} = \begin{bmatrix} 1-\phi(\mathbf{x}_i'\boldsymbol{\beta}_2) \\ (1-\beta_1)\ \phi(\mathbf{x}_i'\boldsymbol{\beta}_2)\ \mathbf{x}_i \end{bmatrix}$$

The weights are

$$(A6) \quad w_i = \frac{r_i}{\pi_i(1-\pi_i)}$$

$$(A7) \quad w_i = \frac{y_i}{\pi_i^2} + \frac{r_i-y_i}{(1-\pi_i)^2}$$

and the "responses"

$$z_i = \left\{ \frac{y_i}{\pi_i} - \frac{r_i-y_i}{1-\pi_i} \right\} / w_i \quad .$$

Dividing the i'th row of $\mathbf{X}$ and z_i by $(1-\beta_1)\phi(\mathbf{x}_i'\boldsymbol{\beta}_2)$ and multiplying the corresponding weight by its square for i=1,...,n, means that only one column of $\mathbf{X}$ needs to be recalculated in each iteration.

A preliminary estimate of β_1 can often be obtained from the proportion of successes in a control group with $\pi_i = \beta_1$ for this type of model. If β_1 is held fixed at that value, the model has a linear systematic part $\nu_i = \mathbf{x}_i'\boldsymbol{\beta}_2$. From the saturated model of this type the maximum likelihood estimate of ν_i would be

$$\phi^{-1}(\max(0 , (y_i/r_i-\beta_1)/(1-\beta_1)))$$

with a minor adjustment of the argument of $\phi^{-1}(\cdot)$ if that would be 0 or 1. These values of $\boldsymbol{\nu}$ and β_1 can be used in the first iteration of NRL.

To illustrate these two NRL algorithms, they were applied to the data in Table 3.5.7 with $\mathbf{x}_i'\boldsymbol{\beta}_2 = \beta_2+\beta_3 d_i$ and $\phi(\cdot)$ defined to be the

TABLE 3.5.5

Iterations of Three NRL Algorithms Fitting a Probit Model to 11 Randomly Generated Binomial Values with r_i=100.

Starting Values $\beta_1 = \Phi^{-1}(0.4)$ and $\beta_2 = 0$.

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C
Algorithm (A3) - NR				
0	-0.25335	0.00000		
1	-0.10619	0.25281	-27.98237	0.31320
2	-0.05144	0.34556	-3.58256	0.19138
3	-0.03803	0.36561	-2.82093	0.04141
4	-0.03743	0.36646	-2.81968	0.00173
5	-0.03743	0.36646	-2.81968	0.00000
Algorithm (A4) - FS				
0	-0.25335	0.00000		
1	-0.06134	0.26684	-20.14835	0.21211
2	-0.04767	0.34439	-3.62710	0.26154
3	-0.03843	0.36502	-2.82322	0.07547
4	-0.03745	0.36645	-2.81968	0.01203
5	-0.03743	0.36646	-2.81968	0.00483
6	-0.03743	0.36646	-2.81968	0.00471
Algorithm (A5)				
0	-0.25335	0.00000		
1	-0.23703	0.21931	-60.26951	0.59540
2	0.00223	0.42760	-8.64868	0.29072
3	-0.02991	0.38047	-3.13579	0.21367
4	-0.03765	0.36709	-2.82027	0.03943
5	-0.03747	0.36646	-2.81968	0.04367
6	-0.03743	0.36646	-2.81968	0.00953
7	-0.03743	0.36646	-2.81968	0.05954

TABLE 3.5.6

Iterations of Three NRL Algorithms Fitting a Probit Model to 11 Randomly Generated Binomial Values with r_i=1000.

Starting Values $\beta_1 = \Phi^{-1}(0.4)$ and $\beta_2 = 0$.

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C
Algorithm (A3) - NR				
0	-0.25335	0.00000		
1	-0.05646	0.25656	-353.49978	0.35773
2	0.02579	0.36412	-20.26397	0.24846
3	0.05215	0.39601	-2.68661	0.07175
4	0.05420	0.39844	-2.60086	0.00525
5	0.05421	0.39845	-2.60086	0.00000
Algorithm (A4) - FS				
0	-0.25335	0.00000		
1	-0.00799	0.27209	-253.43235	0.26709
2	0.03039	0.36260	-21.04998	0.31648
3	0.05140	0.39465	-2.80299	0.11085
4	0.05418	0.39840	-2.60089	0.01324
5	0.05421	0.39845	-2.60086	0.00161
6	0.05421	0.39845	-2.60086	0.00332
Algorithm (A5)				
0	-0.25335	0.00000		
1	-0.19384	0.21934	-743.63197	0.60504
2	0.10271	0.48139	-85.21790	0.30770
3	0.07215	0.42312	-10.75191	0.32413
4	0.05641	0.40088	-2.68647	0.10863
5	0.05425	0.39847	-2.60086	0.01257
6	0.05421	0.39845	-2.60086	0.00258
7	0.05421	0.39845	-2.60086	0.00471

logistic distribution function, $\Phi(y) = \exp(y)/(1+\exp(y))$; the iterations are shown in Table 3.5.8. The iterations of the NR algorithm are also shown for comparison. On convergence,

$$(\mathbf{X'WX-B}) = \begin{Bmatrix} 134.789440 & & \\ 14.900531 & 4.546459 & \\ 44.793342 & 16.637973 & 65.520109 \end{Bmatrix}$$

TABLE 3.5.7
Artificial Data for Fitting Residual Responses Logit Model

Number of Trials, r_i	Number of Successes, y_i	Explanatory Variable, d_i
4	1	Controls ($d_i = -\infty$)
10	3	1
10	2	2
10	5	3
10	8	4
10	8	5

$$(A6)\ \mathbf{B} = \begin{bmatrix} 2.189214 & & \\ -0.464569 & 0.098792 & \\ -0.418784 & 0.089321 & -0.360610 \end{bmatrix}$$

$$(A7)\ \mathbf{B} = \begin{bmatrix} 0.000412 & & \\ -0.000256 & 0.038226 & \\ -0.000300 & -0.733456 & -6.180223 \end{bmatrix}$$

Note that for algorithm (A7) the systematic part is linear in β_1 and so the elements of **B** relating to β_1 are small; if the iterations were continued, these elements would become exactly zero. The systematic part in algorithm (A6) is more nearly linear in β_3 than that in algorithm (A7). Although both matrices **B** are small compared to $(\mathbf{X}'\mathbf{W}\mathbf{X}-\mathbf{B})$, the latter is nearly singular, so the inverse matrices are

$$(NR)\ (\mathbf{X}'\mathbf{W}\mathbf{X}-\mathbf{B})^{-1} = \begin{bmatrix} 0.015269 & & \\ -0.167467 & 4.947570 & \\ 0.032088 & -1.141881 & 0.283291 \end{bmatrix}$$

TABLE 3.5.8

Iterations of Two NRL Algorithms Applied to Data in Table 3.5.7. Starting Values (Iteration 0) were Found as Described in the Text.

Iteration	β_1	β_2	β_3	Log-Likelihood	Convergence Rate, C
Algorithm (A6) - FS					
0	0.14546	-2.29687	0.74910	-1.46750	
1	0.25874	-3.83732	1.05606	-1.07271	0.15860
2	0.20910	-3.99080	1.11448	-0.91757	0.27507
3	0.21446	-4.10861	1.14004	-0.91715	0.53582
4	0.21102	-4.04137	1.12581	-0.91702	0.63657
5	0.21304	-4.08288	1.13464	-0.91697	0.58551
6	0.21180	-4.05815	1.12939	-0.91695	0.61455
7	0.21254	-4.07319	1.13259	-0.91695	0.59682
8	0.21209	-4.06415	1.13067	-0.91694	0.60743
9	0.21236	-4.06962	1.13183	-0.91694	0.60100
10	0.21220	-4.06633	1.13113	-0.91694	0.60486
11	0.21230	-4.06831	1.13155	-0.91694	0.60253
12	0.21224	-4.06711	1.13130	-0.91694	0.60394
13	0.21227	-4.06784	1.13145	-0.91694	0.60308
14	0.21225	-4.06740	1.13136	-0.91694	0.60360
15	0.21226	-4.06767	1.13142	-0.91694	0.60329
Algorithm (A7)					
0	0.14434	-2.29175	0.74842	-1.47056	
1	0.21820	-3.60561	1.01263	-0.96635	0.26346
2	0.22090	-4.18146	1.14914	-0.92031	0.23912
3	0.20895	-3.97710	1.11218	-0.91786	0.80547
4	0.21418	-4.10656	1.13898	-0.91715	0.42923
5	0.21136	-4.04592	1.12699	-0.91700	0.55570
6	0.21273	-4.07817	1.13351	-0.91696	0.48958
7	0.21202	-4.06205	1.13028	-0.91695	0.52001
8	0.21238	-4.07035	1.13195	-0.91694	0.50468
9	0.21220	-4.06614	1.13111	-0.91694	0.51221
10	0.21229	-4.06829	1.13154	-0.91694	0.50842
11	0.21224	-4.06720	1.13132	-0.91694	0.51032
12	0.21227	-4.06775	1.13143	-0.91694	0.50936
13	0.21225	-4.06747	1.13138	-0.91694	0.50984
NR Algorithm					
0	0.19335	-3.54775	1.01175	-0.94689	
1	0.21105	-3.99564	1.11520	-0.91761	0.13570
2	0.21220	-4.06550	1.13094	-0.91694	0.02887
3	0.21226	-4.06756	1.13139	-0.91694	0.00072
4	0.21226	-4.06757	1.13140	-0.91694	0.00000

$$(A6)\quad (X'\hat{W}X)^{-1} = \begin{bmatrix} 0.021587 & & \\ -0.297550 & 7.540242 & \\ 0.059927 & -1.689841 & 0.398526 \end{bmatrix}$$

$$(A7)\quad (X'\hat{W}X)^{-1} = \begin{bmatrix} 0.019198 & & \\ -0.259485 & 6.845969 & \\ 0.050874 & -1.496520 & 0.344739 \end{bmatrix}$$

Both NRL algorithms consistently take too small a step and their convergence is very slow. For other sets of data that were tried, the NRL algorithms took even poorer step directions and sizes.

Much care must therefore be taken when using NRL algorithms for small or poorly fitting data sets. If the model being fitted can be written with a linear systematic part, the NRL algorithm defined from that is equivalent to NR and has the best asymptotic convergence rate. If the model has no linear systematic part, then the best asymptotic convergence is found by using as linear as possible a definition of the systematic part. However in some models, all NRL algorithms may be asymptotically slow to converge and another type of algorithm such as NR or a Quasi-Newton (QN) algorithm may be preferred.

This section's discussion of the relative merits of the various NRL algorithms has been based on asymptotic convergence rates near the maximum likelihood estimate. Clearly, an algorithm that exhibits a poor asymptotic convergence rate is unsatisfactory. As will be seen in Section 3.9, it can also lead to poor estimates of the variances and covariances of the maximum likelihood estimators. However, an algorithm with near-quadratic convergence may perform poorly until it gets near the maximum likelihood estimate. Performance in the first few iterations is investigated in Section 3.7.

3.6 THE FS ALGORITHM AND ITS RELATIONSHIP TO NRL

The NR and NRL algorithms are not affected by how the response vector $\mathbf{y}$ is involved in the log-likelihood. They treat $\mathbf{y}$ as a vector of known constants. The FS algorithm uses the expected second derivatives of the log-likelihood and its practicality is dependent on how easily this expectation can be evaluated and used.

FS iterations cannot usually be computed with least squares calculations unless the model being fitted can be written with a systematic part $\{\eta_i\}$ that does not depend on $\mathbf{y}$. If the systematic part is non-random, then $\mathbf{X}$ does not then depend on $\mathbf{y}$ and

$$\boldsymbol{\beta}^{+}_{FS} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{W}^{*}\mathbf{X})^{-1}\mathbf{X}'\mathbf{v} \tag{3.6.1}$$

$$= \boldsymbol{\beta} + (\mathbf{X}'\mathbf{W}^{*}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{*}\mathbf{z} \tag{3.6.2}$$

$$= (\mathbf{X}'\mathbf{W}^{*}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{*}\mathbf{z}^{*} \tag{3.6.3}$$

where $\mathbf{W}^{*} = E[\mathbf{W}]$ is diagonal. The formulae (3.6.2) and (3.6.3) parallel (3.4.1) and (3.4.2) for the NRL algorithm. For most definitions of log-likelihood components, such as when they are log probability (density) functions or log conditional probability (density) functions, the diagonal elements of $\mathbf{W}^{*}$ are non-negative and therefore FS can be implemented with the weighted least squares routines of many existing statistical computer packages.

FS is easiest to implement for generalized linear models and was described by Nelder and Wedderburn (1972) for these models and by Wedderburn (1974) and Jennrich and Moore (1975) for the corresponding models with nonlinear systematic parts. In this chapter, we shall ignore the auxiliary (nuisance) parameter in these models and therefore write their log-likelihood components as

$$\ell_i(\eta_i) = y_i\theta_i - g(\theta_i)$$

where $\theta_i = \theta(\eta_i)$ and $\eta_i = \eta_i(\boldsymbol{\beta})$. The FS algorithm is then identical to the NRL algorithm that treats θ_i as the model's systematic part. Our discussion of the performance of NRL algorithms in Section 3.5 can therefore be used to assess FS. Its asymptotic convergence is likely to be good only if θ_i is nearly linear, if the model is a "perfect" fit to the data or if the data are a large random sample from the model being fitted. For example in Poisson models with mean $E[y_i] = \lambda_i$, FS is identical to NRL applied to the systematic part $\eta_i = \log(\lambda_i)$. The algorithm (A2) in Section 3.4 is therefore identical to FS and since η_i is nonlinear in models with $\lambda_i = \mathbf{x}_i'\boldsymbol{\beta}$, the asymptotic performance of FS can be poor in these models compared to NRL based on systematic part $\eta_i = \lambda_i$. For all models involving binomial distributions with parameters r_i and π_i, FS is equivalent to NRL applied to the systematic part $\log(\pi_i/(1-\pi_i))$. Algorithms (A4) and (A6) in Section 3.5 are therefore identical to FS.

In models outside the exponential family, FS is usually not the same as any NRL algorithm. However $\mathbf{W}^* = E[\mathbf{W}]$ often cannot be easily evaluated in these models and FS is therefore often not practical. We next describe an example outside the class of generalized linear models, but where FS can be used. In censored distributions, the exact failure time y_i is only known provided $y_i < t_i$; otherwise failure time is just recorded as "above t_i". If the probability density function of y_i is $f(y_i \mid \eta_i)$ then the log-likelihood component of the i'th individual can be written as

$$\ell_i(\eta_i) = \begin{cases} \log f(y_i \mid \eta_i) & \text{if } y_i < t_i \\ \log \int_{t_i}^{\infty} f(y_i \mid \eta_i)\,dy_i & \text{if } y_i \geq t_i \end{cases}$$

If the uncensored data has an exponential distribution with $f(y_i, \eta_i) = \exp\{\eta_i - y_i \exp(\eta_i)\}$ and $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$, then the NRL iterations based on linear systematic part η_i are equivalent to NR and have "explanatory" variables $\mathbf{x}_i$, weights

$$w_i = \begin{cases} y_i \exp(\eta_i) & \text{if i'th not censored} \\ t_i \exp(\eta_i) & \text{if i'th censored} \end{cases}$$

and "responses",

$$z_i = \begin{cases} \exp(-\eta_i)/y_i - 1 & \text{if i'th not censored} \\ -1 & \text{if i'th censored} \end{cases}$$

In this example $w_i>0$ for all i and the systematic part is therefore convex. For FS, the weights and "responses" are

$$w_i = \exp(-\eta_i)\,(1 - \exp(-\exp(\eta_i)t_i))$$

and

$$z_i = \begin{cases} (1 - y_i\exp(\eta_i)) \,/\, w_i & \text{if i'th not censored} \\ -t_i\exp(\eta_i) \,/\, w_i & \text{if i'th censored} \end{cases}$$

respectively. The FS formulae have the disadvantage that even for individuals that were not censored, the time they would have been censored must be known. This information may not be available and, even if it is, it can be argued that it should have no bearing on the algorithm used. NRL also has quadratic convergence and therefore seems the better method to use. The NRL iterations are equivalent to those underlying Aitkin and Clayton's(1980) method which was described at the end of Section 3.1.

In models that can be expressed with a non-random systematic part and that are not in an exponential family, the results in Section 3.5 cannot be directly used to examine the asymptotic convergence of the FS algorithm. However, one similar result holds: the FS algorithm does not usually have quadratic convergence, but its performance becomes close to that of NR if there if a large amount of data that fits the model. It does not always however have quadratic convergence if applied to a small data set that the model fits perfectly.

3.7 IMPLEMENTATION OF ITERATIVELY REWEIGHTED LEAST SQUARES ALGORITHMS IN THE INITIAL ITERATIONS

In Section 3.5, it was shown that an algorithm whose performance is close to NR should be used near the maximum likelihood estimate to get good asymptotic convergence. In this thesis, we are largely concerned with algorithms that can be implemented with iteratively reweighted least squares (IRLS) and, of these, a NRL algorithm based on as linear as possible a systematic part is usually best. In the initial iterations however, we shall show that other IRLS algorithms can perform considerably better than this type of algorithm.

In most problems, the best iteration of the form

$$\boldsymbol{\beta}^{+} = \boldsymbol{\beta} - \mathbf{A}^{-1}\mathbf{g}$$

where $\mathbf{g}$ is the gradient at $\boldsymbol{\beta}$, would use a matrix $\mathbf{A}$ that is intermediate between the second derivative matrices of the log-likelihood at $\boldsymbol{\beta}$ and at the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$. In problems where the second derivative matrix changes markedly when $\boldsymbol{\beta}$ moves away from $\hat{\boldsymbol{\beta}}$, any NRL algorithm where $-(\mathbf{X}'\mathbf{W}\mathbf{X})$ is close to the second derivative matrix at $\boldsymbol{\beta}$ can therefore be far from optimal in the initial iterations. The most extreme effect of this type is when the second derivative matrix is not negative definite at $\boldsymbol{\beta}$, so that the unmodified NRL step would not even be in a descent direction. In other cases where the curvature is much greater at $\boldsymbol{\beta}$ than at $\hat{\boldsymbol{\beta}}$, the initial steps are much too small.

For example, if $y_i/(\mathbf{x}_i'\boldsymbol{\beta})$ has a χ^2 distribution with 1 degree of freedom for $i=1,\ldots,n$ then the NR iterations, which are identical to NRL with systematic part $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$, can be evaluated as least squares calculations with "explanatory" variables $\mathbf{x}_i$, weights

$$w_i = \left(\frac{y_i}{\mathbf{x}_i'\boldsymbol{\beta}} - \frac{1}{2} \right) / (\mathbf{x}_i'\boldsymbol{\beta})^2$$

and "responses"

$$z_i = \left(\frac{y_i}{\mathbf{x}_i'\boldsymbol{\beta}} - 1 \right) / (2w_i\mathbf{x}_i'\boldsymbol{\beta})$$

Starting values can be found by an iteration with $\mathbf{x}_i'\boldsymbol{\beta}$ replaced by y_i, the maximum likelihood estimate of η_i from the saturated model, as described in Section 3.4. In this case, the "responses" would be $z_i^* = y_i$. When applied to the artificial data with $\mathbf{x}_i'\boldsymbol{\beta} = \beta_1+\beta_2 d_i$ in Table 3.7.1, the NR iterations are as shown in Table 3.7.2. Though the asymptotic convergence rate is good, the initial iterations assume too high a curvature and are too short. If the iterations are started from the maximum likelihood estimate in the model with $\beta_2 = 0$, $\hat{\beta}_1 = \sum y_i/n$, the second derivative matrix is not positive definite and the unmodified NR algorithm does not converge.

In contrast, the FS algorithm, which is identical to NRL based on systematic part $\eta_i = (\mathbf{x}_i'\boldsymbol{\beta})^{-1}$, is equivalent to IRLS calculations of the same form as those for NR, but with weights

$$w_i = (\mathbf{x}_i'\boldsymbol{\beta})^{-2}$$

Its iterations from the two different starting values are shown in Table 3.7.3. Since its weights depend on $(\mathbf{x}_i'\boldsymbol{\beta})^{-2}$ rather than $(\mathbf{x}_i'\boldsymbol{\beta})^{-3}$ when $(\mathbf{x}_i'\boldsymbol{\beta})$ is small, it is less extremely affected by poor starting values and its initial iterations are much better than those of the NR algorithm. Since its systematic part is convex, it never needs definiteness modifications. Its asymptotic convergence rate is also good in this example, so it would be preferred to NR.

The most stable IRLS algorithm in the initial iterations is therefore one with positive weights that do not change too sharply as $\boldsymbol{\beta}$ moves from $\hat{\boldsymbol{\beta}}$. In models that can be written with a non-random systematic part, the FS weights are often good to use if they can be easily evaluated. In other circumstances where the log-likelihood components are each bounded above, the NRL algorithm based on (3.3.8) is often better than NR in the initial iterations since its weights are always positive. If such an algorithm exists, we therefore recommend a combination of it and a NRL algorithm that behaves like NR, with a

TABLE 3.7.1

Artificial Data for Fitting Gamma Model

Response, y_i	Explanatory Variable, d_i
0.3153	1
0.1923	1
5.0425	2
0.5692	2
3.7922	3
1.1081	3
8.1781	4
11.110	5

TABLE 3.7.2

Iterations of the NR Algorithm Applied to Data in Table 3.7.1. Starting Values (Iteration 0) were Found from Saturated Model

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C
0	-0.35333	0.56912	-10.22655	
1	-0.58180	0.83101	-8.42021	0.83806
2	-0.91704	1.17564	-7.51031	0.73209
3	-1.30566	1.56178	-7.12938	0.58297
4	-1.63808	1.89310	-7.02035	0.38716
5	-1.81215	2.06693	-7.00599	0.17033
6	-1.84676	2.10151	-7.00560	0.03123
7	-1.84787	2.10263	-7.00560	0.00099
8	-1.84788	2.10263	-7.00560	0.00000

TABLE 3.7.3

Iterations of the FS Algorithm Applied to Data in Table 3.7.1 with Starting Values (Iteration 0) from Two Simpler Models. (Bracketed iterations had stepsize modifications applied)

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C
Starting values from saturated model				
0	-0.35333	0.56912	-10.22655	
1	-1.84165	2.09815	-7.00563	0.00354
2	-1.84782	2.10258	-7.00560	0.00990
3	-1.84788	2.10263	-7.00560	0.00819
Starting values from model with $\beta_2 = 0$				
0	3.78846	0.00000	-9.32784	
1	(-2.85900	2.53237	- ∞)	
	2.12660	0.63309	-8.68694	0.70345
2	(-2.24830	2.28747	-10.62268)	
	1.03287	1.04669	-8.32101	0.72312
3	-2.00690	2.18065	-7.08906	0.06021
4	-1.85029	2.10458	-7.00561	0.01841
5	-1.84789	2.10264	-7.00560	0.00748
6	-1.84788	2.10263	-7.00560	0.00817

switch being made between the algorithms used if the convergence rate of the one in use seems slow. This strategy has the advantage that there is no overhead in applications where both IRLS algorithms have good convergence, but it reduces the risk of slow initial or asymptotic convergence and avoids the need to apply definiteness modifications. Details of when to switch between algorithms are discussed later in this section.

If an IRLS algorithm with positive weights cannot be found for a particular model, then a standard definiteness modification must be used. For example, the subroutine COMBINEM in Appendix B has the same effect as subroutine COMBINE which was described in Section 3.4, but performs a Gill and Murray (1974) definiteness modification rather than returning an error code if ($\mathbf{X}'\mathbf{W}\mathbf{X}$) is not positive definite. This is

done by adding a diagonal matrix $\mathbf{E}$ to $\mathbf{X'WX}$ to make its Choleski factorization stable. COMBINEM uses similar Givens rotations to those used by COMBINE to find upper triangular $\mathbf{R}$ and diagonal $\mathbf{D}$ such that $\mathbf{R'DR} = \mathbf{R}^{*\prime}\mathbf{D}^{*}\mathbf{R}^{*}$ where

$$\mathbf{R}^{*} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{I} : \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{D}^{*} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{D}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E} \end{bmatrix} .$$

and $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{D}_1$ and $\mathbf{D}_2$ are as defined in Section 3.4. $\mathbf{R}$ and $\mathbf{D}$ are used in an identical way to the corresponding matrices returned by COMBINE. Since COMBINEM always calculates the maximum diagonal and off-diagonal elements of $\mathbf{X'WX}$ even when $\mathbf{X'WX}$ is positive definite, this overhead can be avoided if COMBINEM is only called after COMBINE returns an error code.

Even though the modified algorithm is always in a descent direction, $(\mathbf{X'WX}+\mathbf{E})$ is often nearly singular after a modification is made and it can therefore take an unreasonably large step. This can be avoided by finally increasing the diagonal of $(\mathbf{X'WX}+\mathbf{E})$ by ten percent if a definiteness modification has been made. Subroutine DIMULT in Appendix B can be used to do this.

Other definiteness modifications are possible, but are not examined further in this thesis. For example, the negative weights in $\mathbf{D}_2$ could be reduced by a proportion δ until $\mathbf{R'DR} = \mathbf{R}_1'\mathbf{D}_1\mathbf{R}_1 - \delta\mathbf{R}_2'\mathbf{D}_2\mathbf{R}_2$ becomes positive semi-definite.

All NRL and FS algorithms may also require stepsize modifications in the initial iterations. Marquardt-type modifications involve additions to the diagonal of $(\mathbf{X'WX})$ until the log-likelihood is increased. Stirling(1984) showed that each addition of a diagonal $\mathbf{F}$ to $(\mathbf{X'WX})$ is equivalent to p extra rows of data $[\mathbf{I} : \mathbf{0}]$ with weights given by the diagonal of $\mathbf{F}$. Givens rotations can then be used to make the appropriate modifications to $\mathbf{R}$ and $\mathbf{D}$. The diagonal of $\mathbf{F}$ is usually chosen to be proportional to the diagonal of $\mathbf{X'WX}$. Stirling(1984) showed that this type of modification can also be used to apply a

definiteness modification. If $\mathbf{G}$ is a diagonal matrix with positive diagonal elements, then there is some $k>0$ such that $(\mathbf{X}'\mathbf{W}\mathbf{X} + k\mathbf{G})$ is positive definite and

$$\boldsymbol{\beta}^k = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{W}\mathbf{X} + k\mathbf{G})^{-1}\mathbf{X}'\mathbf{W}\mathbf{z}$$

increases the log-likelihood over that of $\boldsymbol{\beta}$. The scalar k can be increased until both conditions are met. Setting the diagonal of $\mathbf{G}$ to the diagonal of $(\mathbf{X}'\mathbf{W}\mathbf{X})$ is usually suitable if its diagonal elements are all positive; this however cannot be guaranteed if $\boldsymbol{\beta}$ is not close to the maximum likelihood estimate and there is no obvious choice of $\mathbf{G}$ if a diagonal element of $(\mathbf{X}'\mathbf{W}\mathbf{X})$ is zero. The method however seems adequate in many problems.

In this thesis however, the simpler stepsize algorithm suggested by Bard (1974, pages 110-113) will be used. If the NRL step direction (with definiteness modification) is given by (3.4.4), then iterations of the form,

$$\boldsymbol{\beta}^+(\delta) = \boldsymbol{\beta} + \delta\mathbf{R}_1^{-1}\mathbf{r}_2$$

are used where the scalar δ denotes the stepsize. The log-likelihood is first evaluated at an initial trial stepsize (often $\delta=1$). Since log-likelihood evaluations with new δ involve computations of order $O(np)$ whereas full iterations involve computations of order $O(np^2)$, it is clearly worth spending some effort in improving the initial stepsize when p is large, even if the initial stepsize increases the log-likelihood. Since

$$\left\{ \frac{\partial \ell(\boldsymbol{\beta}^+(\delta))}{\partial \delta} \right\}_{\delta=0} = \mathbf{r}_2'\mathbf{r}_2$$

and the log-likelihoods at $\delta=0$ and the initial trial stepsize are known, Bard's algorithm estimates the optimum stepsize δ to maximize a fitted quadratic in δ defined by these three quantities. If the initial stepsize reduces the log-likelihood, interpolation is done in this way. Otherwise, a similar interpolation or extrapolation is tried if the estimate of the optimum stepsize is less than a proportion $(1-\rho)$ or more than $(1-\rho)^{-1}$ different from the initial stepsize. Setting ρ to

0.1 is usually suitable. If ρ=1.0, no interpolation or extrapolation is performed unless the initial stepsize reduced the log-likelihood. Subroutine STEP in Appendix B performs this stepsize modification and its input parameter PACC defines the value of ρ. The initial trial stepsize can be based on the stepsize used in the previous iteration, but in the numerical examples that use STEP in this thesis, an initial unit stepsize is always used.

This type of stepsize algorithm can also be used to assess the convergence rate of the algorithm being used and, in situations where there are two IRLS algorithms available, an interpolated or extrapolated stepsize of, say δ<0.8 or δ>1.25, can be used to trigger a switch to the other algorithm. Since the examples in later sections of this thesis are intended to allow comparison of different IRLS definitions of weights, switching between algorithms is not implemented here and the parameter ρ of STEP is usually set to 1.0.

3.8 CONSTRAINTS

In this section we consider fitting models subject to constraints of the form $\mathbf{C}(\boldsymbol{\beta}) = \mathbf{c}$ where $\mathbf{C}(\cdot)$ is a vector function and $\mathbf{c}$ is a vector defining the q constraints.

If $\mathbf{c}$ was a normally distributed prior estimate of $\mathbf{C}(\boldsymbol{\beta})$ with $\text{var}(\mathbf{c}) = k^{-1}\mathbf{I}$, then the overall log-likelihood would be $\sum_{i=1}^{n+q} \ell_i(\boldsymbol{\beta})$ where

$$\ell_i(\boldsymbol{\beta}) = -\frac{k\,(c_i - C_i(\boldsymbol{\beta}))^2}{2} \qquad \text{for } i=(n+1), \ldots, (n+q)$$

FS and NRL are both equivalent to adding "response" and "explanatory" variables,

$$\begin{bmatrix} z_{n+1} \\ \vdots \\ z_{n+q} \end{bmatrix} = \mathbf{c} - \mathbf{C}(\boldsymbol{\beta})$$

and

$$\begin{bmatrix} \mathbf{x}_{n+1} \\ \vdots \\ \mathbf{x}_{n+q} \end{bmatrix} = \mathbf{C}'(\boldsymbol{\beta})$$

to those used for fitting the unconstrained model in question, each with weight k. The fully imposed constraints are equivalent to letting $k \to \infty$ in each iteration.

This method is equivalent to linearizing the constraints

$$\mathbf{c} = \mathbf{C}(\boldsymbol{\beta}^+) \approx \mathbf{C}(\boldsymbol{\beta}) + (\boldsymbol{\beta}^+ - \boldsymbol{\beta})\,\mathbf{C}'(\boldsymbol{\beta})$$

and applying these exactly as linear constraints in all iterations. As described in Section 2.5, the Givens algorithm for least squares can be easily modified to accept such constraints.

If the algorithm converges, then clearly $\mathbf{C}(\hat{\boldsymbol{\beta}}) = \mathbf{c}$ and the constraints are satisfied. Also by letting $k \to \infty$ it can be shown that

$\sum_{i=n+1}^{n+q} \ell_i(\hat{\boldsymbol{\beta}}) = 0$ and so $\hat{\boldsymbol{\beta}}$ is a local maximum of the log-likelihood subject to the constraints.

We have only dealt with equality constraints here. Some models also have inequality constraints. These occasionally relate to a single parameter, such as $\beta_1 > 0$, but more often are constraints on all $\eta_i(\boldsymbol{\beta})$. For example, in Poisson models with $E[y_i] = \eta_i(\boldsymbol{\beta})$, we must have $\eta_i(\boldsymbol{\beta}) \geq 0$ for all i. If the maximum likelihood estimate is not on a boundary, but is at a turning point of the likelihood, then stepsize modifications can be used after the first iteration to keep the parameters in a feasible region of the parameter space; we can therefore treat infeasible values as having a log-likelihood of $-\infty$.

There can however be difficulties in obtaining a feasible starting value for the algorithms. If this problem is encountered or if the maximum likelihood estimate may be on a boundary, a constrained optimization algorithm must be used. The necessary modifications to the IRLS algorithms are not persued in this thesis, but it is noted that they involve applying equality constraints in iterations and these can be done as described above.

3.9 VARIANCES AND TESTS

The standard results about maximum likelihood estimators can be applied to $\hat{\boldsymbol{\beta}}$. It asymptotically has a normal distribution with mean $\boldsymbol{\beta}$. Its variance-covariance matrix can be estimated by $-\mathbf{A}^{-1}$ where $\mathbf{A}$ is an approximation to the Hessian, $\mathbf{H}$. The matrix $\mathbf{A}$ that would be used in this formula is the approximation used in the last iteration of the NR, NRL or FS algorithm which is used to calculate the estimate $\hat{\boldsymbol{\beta}}$. If FS is used, $-\mathbf{A}^{-1}$ is the inverse of Fisher's information matrix and this is the most commonly used type of estimate of the variances of maximum likelihood estimators. However Efron and Hinkley (1978) give reasons for preferring the estimate resulting from the NR algorithm. The matrices can be extracted by subroutine VARS in Appendix B. In algorithms that converge slowly and where the approximation to the second derivative matrix being used is poor, this estimate of the variance can also be very poor. For example, the inverse matrices, $\mathbf{A}^{-1}$, found when applying algorithms (A6) and (A7) to fit a residual responses model to the data in Table 3.5.7, were given in Section 3.5. These both considerably overestimate the variance estimate found from the inverse of the second derivative matrix itself, which is also given in Section 3.5. Similarly, the estimated variances from the FS algorithm applied to the two data sets in Table 3.4.1are very much lower than those from the NR algorithm.

Tests about $\boldsymbol{\beta}$ can be based on the asymptotic properties of the generalized likelihood ratio test. The deviance of a model can be defined as $2(L_0-\Sigma\ell_i(\hat{\eta}_i))$ where L_0 is the maximum of $\Sigma\ell_i(\eta_i)$ over all choices of η_i (that is, from the saturated model). A model's deviance can be used to describe its fit, and can be partitioned for a nested sequence of models and presented in a table analogous to an analysis of variance table. The difference in deviances between different nested models will asymptotically have a χ^2 distribution with degrees of freedom equal to the number of parameters in one model restricted by

the other. The results and definitions here are an extension of those of Nelder and Wedderburn(1972) for generalized linear models.

TABLE 3.9.1
Numbers of Insects Affected by Various Concentrations of Two Types of Derris Roots (Martin, 1940)

Type of Derris Root Used	Dry Root mg./l.	No of Insects Used	Number Affected
W.213	1480	142	142
	1000	127	126
	480	128	115
	120	126	58
W.214	619	125	125
	458	117	115
	310	127	114
	149	51	40
	37.1	132	37
Controls	---	129	21

For example, the data in Table 3.9.1 report the numbers of a grain beetle that were affected by different concentrations of two derris roots W.213 and W.214. There were also 129 control insects that were sprayed only with the medium in which the derris was mixed, but no derris. A logit residual responses model was used with

$$\pi_i = \beta_1+(1-\beta_1)\Phi(\beta_2+\beta_4(\log \text{concentration}))$$
for insects getting root W.213,
$$\pi_i = \beta_1+(1-\beta_1)\Phi(\beta_3+\beta_5(\log \text{concentration})) \qquad (3.9.1)$$
for insects getting root W.214, and
$\pi_i = \beta_1$ for insects getting no derris.

The control insects therefore correspond to zero concentration. The function $\Phi(\cdot)$ was taken to be the logistic cumulative distribution function, $\Phi(y) = \exp(y)/(1+\exp(y))$. The model was fitted using the NRL algorithm (A6) from Section 3.5, and was also fitted with the constraints, $\beta_4 = \beta_5$ and with the two constraints, $\beta_2 = \beta_3$ and $\beta_4 = \beta_5$.

TABLE 3.9.2
Analysis of Deviance Table for Residual Responses Logit Models Applied to Data in Table 3.9.1.

Model	D.F.	Deviance	Difference in Deviance	D.F.	Description
(3.9.1) with $\beta_4=\beta_5$ and $\beta_2=\beta_3$	7	28.81			
			17.15	1	Equivalence of Roots Assuming Parallelism
(3.9.1) with $\beta_4=\beta_5$	6	11.66			
			0.03	1	Parallelism
(3.9.1)	5	11.63			
			11.63	5	Residual
Saturated Model	0	0			

The deviances from these models are presented in the analysis of deviance table, Table 3.9.2. The residual deviance gives an indication of the adequacy of the full model and its value of 11.63 is just significantat the 5 per cent level when compared to $\chi^2(5 \text{ d.f.})$, giving some evidence that the model may not be appropriate for the data. The difference in deviances corresponding to the constraint $\beta_4 = \beta_5$, 0.03,

is not significant when compared to χ^2(1 d.f.) which gives no evidence that the two derris roots do not have parallel effect. The difference in deviances corresponding to imposing the constraint $\beta_2 = \beta_3$, 17.15, is highly significant when compared to χ^2(1 d.f.), giving strong evidence that the two roots have different potencies. The matrix $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ from the last iteration of the model with $\beta_4 = \beta_5$ gives a consistent estimate of the variance-covariance matrix of the parameter estimates and is

$$\text{var}\begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \\ \hat{\beta}_4 \end{pmatrix} = \begin{bmatrix} 0.000973 & & & \\ -0.015615 & 1.03734 & & \\ 0.002388 & -0.176467 & 0.031231 & \\ 0.000090 & -0.067925 & 0.006090 & 0.076065 \end{bmatrix}$$

Since the asymptotic convergence rate when fitting the model was 0.15, this variance estimate is likely to be acceptably close to that expected from using the NR algorithm.

4. MODELS WITH TWO OR MORE SYSTEMATIC PARTS

4.1 FITTING MODELS WITH ITERATIVELY REWEIGHTED LEAST SQUARES

In many models, the log-likelihood components depend on the parameters essentially through two functions of the parameters and these are often non-random. Though many models, such as those involving the binomial, Poisson, exponential and multinomial distributions, can be written with a single non-random systematic part, there are many other distributions such as the normal, Weibull and negative binomial distributions which depend on two or more functions of parameters and cannot be written with a single systematic part in a way that leads to a NRL algorithm with reasonable convergence. Most models described in Section 3.2 are of this form.

In this section, we therefore consider models with two or more systematic parts and develop a generalization of the NRL algorithm that can be implemented with a sequence of weighted least squares calculations. We first treat models with two systematic parts and log-likelihood components of the form

$$\ell_i(\eta_i^{(1)}, \eta_i^{(2)})$$

where $\eta_i^{(1)} = \eta_i^{(1)}(\boldsymbol{\beta})$ and $\eta_i^{(2)} = \eta_i^{(2)}(\boldsymbol{\beta})$ for $i=1,\ldots,n$. In many, but not all, models the systematic parts are non-random, and in many models the second systematic part will simply be a single auxiliary parameter which is distinct from those involved in $\eta_i^{(1)}$. The class however also includes models which explain variation in two parameters of the response distribution using separate functions of explanatory variables. For example Stirling(1985) describes models where the variance in a normal linear model depends on a different function of explanatory variables from that affecting the mean. Manton and

Woodbury(1981) used a negative binomial model where the mean and shape parameter were affected by different explanatory variables. Glaser(1984) used a Weibull model to describe the lifetimes of certain items and modelled the two Weibull parameters with different functions of explanatory variables.

We can define the NRL algorithm for models with 2 systematic parts in a similar way to the definition used for that algorithm in Section 3.3, by defining $\eta_i^{(1)*}$ and $\eta_i^{(2)*}$ to be the linear terms of a Taylor series for $\eta_i^{(1)}$ and $\eta_i^{(2)}$ round $\boldsymbol{\beta}$. If $\ell^*(\boldsymbol{\beta})$ denotes the log-likelihood with $\eta_i^{(1)}$ and $\eta_i^{(2)}$ replaced by $\eta_i^{(1)*}$ and $\eta_i^{(2)*}$, then the iterations of the NRL algorithm can be defined to be NR iterations applied to $\ell^*(\boldsymbol{\beta})$. As before, $\ell^*(\cdot)$ and $\ell(\cdot)$ have the same first derivative and so converge to the same value. If both systematic parts are linear, then NRL = NR. The NRL iterations can be expressed in the form

$$\boldsymbol{\beta}^{+}_{NRL} = \boldsymbol{\beta} + \left(\begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \\ \mathbf{X}^{(3)} \end{bmatrix}' \begin{bmatrix} \mathbf{W}^{(1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W}^{(3)} \end{bmatrix} \begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \\ \mathbf{X}^{(3)} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \\ \mathbf{X}^{(3)} \end{bmatrix}' \begin{bmatrix} \mathbf{v}^{(1)} \\ \mathbf{v}^{(2)} \\ \mathbf{0} \end{bmatrix} \tag{4.1.1}$$

where $\mathbf{X}^{(j)}$ has i'th row $\mathbf{x}_i^{(j)}{}'$, $\mathbf{W}^{(j)}$ is diagonal with i'th diagonal element $w_i^{(j)}$ and $\mathbf{v}^{(j)}$ is a vector with i'th element $v_i^{(j)}$ for j=1,...,3 and i=1,...,n, with

$$\mathbf{x}_i^{(j)} = \frac{\partial \eta_i^{(j)}}{\partial \boldsymbol{\beta}} \quad \text{for } j=1,2$$

$$\mathbf{x}_i^{(3)} = \mathbf{x}_i^{(1)} + \mathbf{x}_i^{(2)}$$

$$w_i^{(3)} = -\frac{\partial^2 \ell_i}{\partial \eta_i^{(1)}\, \partial \eta_i^{(2)}}$$

$$w_i^{(j)} = -\frac{\partial^2 \ell_i}{\partial \eta_i^{(j)\,2}} - w_i^{(3)} \quad \text{for } j=1,2$$

and

$$v_i^{(j)} = \frac{\partial \ell_i}{\partial \eta_i^{(j)}} \quad \text{for } j=1,2 \ .$$

This is in a similar form to (3.3.5) for models with a single systematic part and can be evaluated as a weighted least squares calculation in a similar way, the main difference being that each of the original observations generates <u>three</u> rows of data and weights.

The main applications of this algorithm are to models which cannot be written with a single non-random systematic part. However, some models with a single nonlinear systematic part can be rewritten with two linear systematic parts and this allows NR to be implemented as IRLS, giving faster convergence in some examples where NRL and FS applied to the single nonlinear systematic part are slow to converge. An important class of models with this type of property is the class of generalized linear models with parametric link functions (Pregibon, 1980 and Scallon et al, 1984). These are of the form (3.1.1) but with $\theta_i = k(\eta_i, \boldsymbol{\beta}_1)$ and $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}_2$. This class of models includes the binomial residual responses model (3.5.4) as well as other ad hoc models described by Pregibon (1980) and Scallon et al (1984). We shall see later in this section that the auxiliary (nuisance) parameter ϕ can be ignored when estimating $\boldsymbol{\beta}$. Either θ_i can be treated as a single systematic part and NRL or FS used, or η_i and $\boldsymbol{\beta}_1$ can be treated as different systematic parts and (4.1.1) used to implement NR. The latter method has quadratic convergence as opposed to the linear convergence of FS and NRL and so would be expected to perform better, at least near the maximum likelihood estimate. The NR iterations in Table 3.5.8 were found using this IRLS method and are a clear improvement over IRLS methods (A6) and (A7).

If a model's two systematic parts are non-random, then the FS iterations can be similarly written as

$$\boldsymbol{\beta}^{+}_{FS} = \boldsymbol{\beta} + \left(\begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \\ \mathbf{X}^{(3)} \end{bmatrix}' \begin{bmatrix} \mathbf{W}^{(1)*} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}^{(2)*} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W}^{(3)*} \end{bmatrix} \begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \\ \mathbf{X}^{(3)} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \\ \mathbf{X}^{(3)} \end{bmatrix}' \begin{bmatrix} \mathbf{v}^{(1)} \\ \mathbf{v}^{(2)} \\ \mathbf{0} \end{bmatrix} \tag{4.1.2}$$

where $\mathbf{W}^{(j)*}$ is diagonal with i'th element $w_i^{(j)*}$ and

$$w_i^{(3)*} = -E\left\{ \frac{\partial^2 \ell_i}{\partial \eta_i^{(1)} \, \partial \eta_i^{(2)}} \right\}$$

$$w_i^{(j)*} = -E\left\{ \frac{\partial^2 \ell_i}{\partial \eta_i^{(j)\,2}} \right\} - w_i^{(3)*} \quad \text{for } j=1,2 \ .$$

This can again be implemented with weighted least squares calculations.

The FS algorithm can be simplified in three special cases :-

Separable systematic parts

If

$$E\left[\frac{\partial^2 \ell_i(\eta_i^{(1)},\eta_i^{(2)})}{\partial \eta_i^{(1)}\,\partial \eta_i^{(2)}}\right] = 0$$

then the two systematic parts are called separable. Then

$$\boldsymbol{\beta}_{FS}^{+} = \boldsymbol{\beta} + \left(\begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \end{bmatrix}' \begin{bmatrix} \mathbf{W}^{(1)*} & 0 \\ 0 & \mathbf{W}^{(2)*} \end{bmatrix} \begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \end{bmatrix}' \begin{bmatrix} \mathbf{v}^{(1)} \\ \mathbf{v}^{(2)} \end{bmatrix} \tag{4.1.3}$$

where $w_i^{(j)*} = -E[\partial^2 \ell_i(\eta_i^{(1)},\eta_i^{(2)})/\partial \eta_i^{(j)\,2}]$ for $j=1,2$ and $i=1,\ldots,n$. The FS algorithm is therefore in a similar form to the FS algorithm with a single systematic part and, if implemented with IRLS, the weights are usually positive.

Detached systematic parts

In the common special case where $\eta_i^{(1)}(\boldsymbol{\beta})$ and $\eta_i^{(2)}(\boldsymbol{\beta})$ are separable and involve completely different subsets of $\boldsymbol{\beta}$, say $\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{\beta}^{(2)}$ respectively, there is a further simplification since

$$\boldsymbol{\beta}_{FS}^{(j)+} = \boldsymbol{\beta}^{(j)} + \left[\, \mathbf{X}^{(j)\prime} \mathbf{W}^{(j)} \mathbf{X}^{(j)} \,\right]^{-1} \mathbf{X}^{(j)\prime} \mathbf{v}^{(j)} \tag{4.1.4}$$

where $\mathbf{X}^{(j)}$ is defined here to have ith row $\mathbf{x}_i^{(j)} = \partial \eta_i^{(j)}/\partial \boldsymbol{\beta}^{(j)}$ for $j=1,2$ and $i=1,\ldots,n$. The FS algorithm can therefore be improved upon by alternating between FS for $\boldsymbol{\beta}^{(1)}$ with $\boldsymbol{\beta}^{(2)}$ held fixed and FS for $\boldsymbol{\beta}^{(2)}$ with $\boldsymbol{\beta}^{(1)}$ held fixed. The two systematic parts are then called detached and the algorithm is called FS-alternation.

Systematic parts that can be factorized out

A final simplification arises when two detached systematic parts are such that $\eta_i^{(1)} = \eta^{(1)}$ for all i and

$$\ell_i(\eta_i^{(1)}, \eta_i^{(2)}) = \eta^{(1)} . k_i(\eta_i^{(2)}) + b_i(\eta^{(1)}) . \qquad (4.1.5)$$

We then say that the systematic part $\eta_i^{(1)}$ can be factorized out. The auxiliary parameter in Nelder and Wedderburn's generalized linear models, which include the normal and gamma distributions, are of this form, as are the auxiliary parameters in Jorgensen's(1983), Pregibon's(1980) and Scallon et al's(1984) generalizations. Clearly the maximum likelihood estimate of $\beta^{(2)}$ for fixed $\beta^{(1)}$ does not depend on the value of $\beta^{(1)}$ and is the overall maximum likelihood estimate of $\beta^{(2)}$. In generalized linear models, the FS iterations for $\beta^{(2)}$ with $\beta^{(1)}$ fixed do not depend on $\beta^{(1)}$ and so the iterations for $\beta^{(2)}$ can all be performed before the iterations for $\beta^{(1)}$ are started. Often, estimates of the auxiliary parameters are not required and the iterations for $\beta^{(1)}$ are omitted. In general however, the FS iterations for $\beta^{(2)}$ with $\beta^{(1)}$ fixed may depend on $\beta^{(1)}$ (Jorgensen, 1983), but an arbitrary sequence of values for $\beta^{(1)}$ may be used in the iterations.

The concepts of separable and detached systematic parts are not directly applicable to the NRL algorithm since NRL does not simplify as in (4.1.3) or (4.1.4) unless $\partial^2 \ell_i(\eta_i^{(1)}, \eta_i^{(2)})/\partial \eta_i^{(1)} \partial \eta_i^{(2)} = 0$ and this rarely holds in models. However in moderate to large samples, the observed and expected second derivative matrices are usually similar and simplifications of the forms (4.1.3) and (4.1.4) will then be almost as fast as joint NRL. (At any rate if the observed and expected Hessians are very different, FS itself will not be an efficient algorithm to use.) If the first systematic part can be factorized out, the NRL iterations for $\beta^{(2)}$ with $\beta^{(1)}$ fixed never depend on $\beta^{(1)}$ and therefore the iterations for $\beta^{(2)}$ can always be done first.

If a model has two systematic parts that are random or are not separable, then simplifications of types (4.1.3) or (4.1.4) to the FS and NRL algorithms can still be used, but the implicit approximation to the Hessian can be poor and so they can sometimes be very slow to

converge. Convergence is more satisfactory if $\partial^2 \ell_i(\eta_i^{(1)}, \eta_i^{(2)})/\partial\eta_i^{(1)}\partial\eta_i^{(2)}$ (or its expected value) is used, as in the IRLS formulae (4.1.1) and (4.1.2).

The concepts and methods described above can be extended to models with 3 or more systematic parts. For example, (4.1.1) would be extended with 3 systematic parts into IRLS calculations where each original observation generates <u>six</u> rows of data. The "explanatory" variables would be $\mathbf{x}_i^{(1)}$, $\mathbf{x}_i^{(2)}$, $\mathbf{x}_i^{(3)}$, $(\mathbf{x}_i^{(1)}+\mathbf{x}_i^{(2)})$, $(\mathbf{x}_i^{(1)}+\mathbf{x}_i^{(3)})$ and $(\mathbf{x}_i^{(2)}+\mathbf{x}_i^{(3)})$, and the "responses" would be $w_i^{(1)-1}v_i^{(1)}$, $w_i^{(2)-1}v_i^{(2)}$, $w_i^{(3)-1}v_i^{(3)}$, 0, 0 and 0; the "weights" would be

$$w_i^{(4)} = -\frac{\partial^2 \ell_i}{\partial\eta_i^{(1)}\,\partial\eta_i^{(2)}} ,$$

similar formulae for $w_i^{(5)}$ and $w_i^{(6)}$,

$$w_i^{(1)} = -\frac{\partial^2 \ell_i}{\partial\eta_i^{(1)\,2}} - w_i^{(4)} - w_i^{(5)} ,$$

and similar formulae for $w_i^{(2)}$ and $w_i^{(3)}$.

If a systematic part is linear, $\mathbf{x}_i^{(j)}$ does not need to be recalculated in each iteration. Convexity of a systematic part is only helpful if the systematic parts are separable or detached; the IRLS weights for NRL are then always positive. In (4.1.1) and (4.1.2), the weights may be negative even if the systematic parts are individually convex.

An alternative approach to implementing NRL is sometimes convenient if there are two systematic parts and the second is a single parameter $\eta_i^{(2)} = \phi$. The NRL iteration (4.1.1) for $\boldsymbol{\beta}' = [\boldsymbol{\beta}^{(1)\prime} : \phi]$ has the form

$$\boldsymbol{\beta}^{+} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{z}$$

where $\mathbf{z} = \mathbf{W}^{-1}\mathbf{v}$ and $\mathbf{X}$, $\mathbf{W}$ and $\mathbf{v}$ are the partitioned matrices in (4.1.1). This is usually evaluated from the Choleski factors $\mathbf{R}$ and $\mathbf{D}$ as described earlier, where

$$\mathbf{R}'\mathbf{D}\mathbf{R} = \begin{bmatrix} \mathbf{X}'\mathbf{W}\mathbf{X} & \mathbf{X}'\mathbf{W}\mathbf{z} \\ \mathbf{z}'\mathbf{W}\mathbf{X} & \mathbf{z}'\mathbf{W}\mathbf{z} \end{bmatrix}$$

and $\mathbf{R}$ and $\mathbf{D}$ can be found in a numerically stable way using a QR algorithm on $\mathbf{X}$,$\mathbf{W}$ and $\mathbf{z}$. However, we can also write

$$\mathbf{R}'\mathbf{D}\mathbf{R} = \begin{bmatrix} \mathbf{X}^{(1)'}\mathbf{W}_1\mathbf{X}^{(1)} & -\dfrac{\partial^2 \ell}{\partial \boldsymbol{\beta}^{(1)} \partial \phi} & \mathbf{X}^{(1)'}\mathbf{W}_1\mathbf{z}^{(1)} \\ -\dfrac{\partial^2 \ell}{\partial \boldsymbol{\beta}^{(1)'} \partial \phi} & -\dfrac{\partial^2 \ell}{\partial \phi^2} & \dfrac{\partial \ell}{\partial \phi} \\ \mathbf{z}^{(1)'}\mathbf{W}_1\mathbf{X}^{(1)} & \dfrac{\partial \ell}{\partial \phi} & \mathbf{z}^{(1)'}\mathbf{W}_1\mathbf{z}^{(1)} \end{bmatrix}$$

The Choleski factorization can also be found in a numerically stable way by augmenting the Choleski factorization

$$\mathbf{R}^{(1)'}\mathbf{D}^{(1)}\mathbf{R}^{(1)} = \begin{bmatrix} \mathbf{X}^{(1)'}\mathbf{W}_1\mathbf{X}^{(1)} & \mathbf{X}^{(1)'}\mathbf{W}_1\mathbf{z}^{(1)} \\ \mathbf{z}^{(1)'}\mathbf{W}_1\mathbf{X}^{(1)} & \mathbf{z}^{(1)'}\mathbf{W}_1\mathbf{z}^{(1)} \end{bmatrix}$$

that can be found using the methods in Chapter 3 for a NRL iteration with ϕ held fixed and therefore one systematic part. The first step is to find the Choleski factorization

$$\mathbf{M}'\mathbf{F}\mathbf{M} = \begin{bmatrix} & & -\dfrac{\partial^2 \ell}{\partial \boldsymbol{\beta}\, \partial \phi} \\ & \mathbf{R}^{(1)'}\mathbf{D}^{(1)}\mathbf{R}^{(1)} & \dfrac{\partial \ell}{\partial \phi} \\ -\dfrac{\partial^2 \ell}{\partial \boldsymbol{\beta}\, \partial \phi} & \dfrac{\partial \ell}{\partial \phi} & -\dfrac{\partial^2 \ell}{\partial \phi^2} \end{bmatrix}$$

The calculations are those to find a new row of a Choleski factorization. Clarke(1981) gave an algorithm to interchange two rows of a Choleski factorization and this method can be used on the last two rows of $\mathbf{M}$ and $\mathbf{F}$ to give $\mathbf{R}$ and $\mathbf{D}$. The step $(\boldsymbol{\beta}^{+}_{NRL} - \boldsymbol{\beta})$ would then be evaluated in the same way as before.

If $\mathbf{X}'\mathbf{W}\mathbf{X}$ is not positive definite, definiteness modifications may be needed both when $\mathbf{R}^{(1)}$ and $\mathbf{D}^{(1)}$ are found and also when $\mathbf{M}$ and $\mathbf{F}$ are found. The former can be applied with the algorithm COMBINEM which was described in Section 3.7; the latter can also be found according to

Gill and Murray's (1974) recommendations. Subroutine NUISNC in Appendix B performs these operations to augment the Choleski factorization of the NRL algorithm with ϕ fixed into that of the NRL algorithm with ϕ not fixed. Further calls of NUISNC can be used to incorporate additional auxiliary parameters. This technique can also be used for the FS algorithm if the second derivatives are replaced by their expected values throughout.

In models with two systematic parts, $\eta_i^{(1)}$ and $\eta_i^{(2)}$ where $\eta_i^{(1)}$ can be factorized out, $\eta_i^{(1)}$ can be fixed at an arbitrary value and NRL or FS can be applied to the remaining systematic part. As this reduces the model to a single systematic part, starting values can be obtained as described in Section 3.4.

To find starting values in other models with two systematic parts, (4.1.1) can be rewritten as

$$\boldsymbol{\beta}^{+}_{NRL} = \left(\begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \\ \mathbf{X}^{(3)} \end{bmatrix}' \begin{bmatrix} & \mathbf{W} & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \\ \mathbf{X}^{(3)} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \\ \mathbf{X}^{(3)} \end{bmatrix}' \begin{bmatrix} & \mathbf{W} & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} \mathbf{z}^{(1)*} \\ \mathbf{z}^{(2)*} \\ \mathbf{z}^{(3)*} \end{bmatrix} \tag{4.1.6}$$

where $\mathbf{W}$ denotes the diagonal matrix of weights, $\mathbf{z}^{(j)*} = (\mathbf{W}^{(j)-1}\mathbf{v}^{(j)} + \mathbf{X}^{(j)}\boldsymbol{\beta})$ for $j=1,2$ and $\mathbf{z}^{(3)*} = \mathbf{X}^{(3)}\boldsymbol{\beta}$. The FS iteration (4.1.2) can be expressed in a similar form. If the two systematic parts are linear, then the least squares calculations expressed in this form only use the estimate from the previous iteration through $\{\eta_i^{(1)}\}$ and $\{\eta_i^{(2)}\}$. If similar models have already been fitted to the data, the systematic parts from the most similar previous model can be used in the calculations to get starting values. If no similar model has been fitted or if the systematic parts are nonlinear, then the user must provide a guess or an alternative estimate for some of the parameters. If the parameters can be split into two subsets, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ and the two systematic parts $\eta_i^{(1)}$ and $\eta_i^{(2)}$ depend only on $\boldsymbol{\beta}_1$ and $(\boldsymbol{\beta}_2'\mathbf{x}_i)$, then so does the least squares calculation (4.1.6). The user need then only specify an initial value for $\boldsymbol{\beta}_1$ and $(\boldsymbol{\beta}_2'\mathbf{x}_i)$ can be replaced in (4.1.6) with the maximum

likelihood estimate from a similar model of this type with fixed β_1, such as a saturated model. If the systematic parts depend on β_1 and $(\beta_2'\mathbf{x}_i(\beta_1))$ then after β_1 has been specified, an iteration of the constrained model with fixed β_1 (and a single linear systematic part) can be used to get a starting value for β_2. These strategies mirror those described for obtaining starting values for models with a single systematic part in Section 3.4.

Similar issues affect convergence and choice of algorithms to those discussed in Chapter 3. To get good asymptotic convergence, an algorithm that behaves similarly to NR must be used. For good performance in the first few iterations, the IRLS weights should not be affected too sharply by changes in β. If one systematic part can be factorized out, then the model can be treated as having only one systematic part and the recommendations in Chapter 3 directly apply. If the systematic parts are separable, the FS weights are positive and FS often performs well in the first few iterations; it can be used in combination with a NRL algorithm that performs like NR. If the systematic parts are not separable, some FS weights can be negative, but $(\mathbf{X}'\mathbf{W}\mathbf{X})$ is always positive definite, so that definiteness modifications are never needed and FS can always be used in the first few iterations. The NRL algorithm based on (3.3.8) often cannot be implemented to get an IRLS algorithm with positive weights if one systematic part cannot be factorized out, since the log-likelihood components are often not bounded above.

4.2 VARIANCES AND TESTS

In models with a single systematic part, the variances of maximum likelihood parameter estimators are estimated by the inverse of the approximation to the Hessian in the iterative algorithm used, and this is the matrix inverted in the last IRLS iteration. A similar result can also be applied to models with two (or more) systematic parts. For models with two (or more) systematic parts, the variances of the maximum likelihood estimators can be estimated by the inverse matrices in (4.1.1) or (4.1.2) which are obtained from the last IRLS iteration, or their extensions to 3 or more systematic parts. For separable or detached systematic parts, the inverse matrices in (4.1.3) or (4.1.4), or the corresponding NRL formulae, can be used. If the systematic parts are detached, the estimators of the two subsets of $\boldsymbol{\beta}$ are asymptotically independent. In models with a systematic part $\eta_i^{(1)}$ that can be factorized out, the IRLS iterations used for $\boldsymbol{\beta}^{(2)}$ usually do not involve $\boldsymbol{\beta}^{(1)}$ and therefore the inverse matrix in the last iteration must be divided by an estimate of $\eta^{(1)}$ to give an estimate of $\text{var}(\hat{\boldsymbol{\beta}}^{(2)})$. In normal linear models, this corresponds to multiplying $(\mathbf{X}'\mathbf{X})^{-1}$ by an estimate of σ^2 to estimate the variance of the least squares estimators.

If a model does not have detached systematic parts, but $\eta_i^{(1)}(\boldsymbol{\beta})$ and $\eta_i^{(2)}(\boldsymbol{\beta})$ involve different subsets of $\boldsymbol{\beta}$, say $\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{\beta}^{(2)}$ respectively, and the alternation algorithm given by (4.1.4) is used, consistent estimates of the variances and covariances of the estimators can still be found. Aitkin and Clayton(1980) showed that they could be estimated by

$$\begin{aligned} \text{var}\,(\hat{\boldsymbol{\beta}}^{(1)}) &= \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}-\mathbf{B}'\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}'\mathbf{A}^{-1} \\ \text{cov}\,(\hat{\boldsymbol{\beta}}^{(2)},\hat{\boldsymbol{\beta}}^{(1)}) &= -\mathbf{A}^{-1}\mathbf{B}(\mathbf{C}-\mathbf{B}'\mathbf{A}^{-1}\mathbf{B})^{-1} \\ \text{var}\,(\hat{\boldsymbol{\beta}}^{(2)}) &= (\mathbf{C}-\mathbf{B}'\mathbf{A}^{-1}\mathbf{B})^{-1} \end{aligned} \qquad (4.2.1)$$

where $\mathbf{A}^{-1} = (\mathbf{X}^{(1)'}\mathbf{W}^{(1)}\mathbf{X}^{(1)})^{-1}$ is calculated in the last IRLS iteration for $\boldsymbol{\beta}^{(1)}$ ($\boldsymbol{\beta}^{(2)}$ fixed), and

$$\mathbf{B} = -E\left\{\frac{\partial^2 \ell}{\partial \boldsymbol{\beta}^{(1)} \partial \boldsymbol{\beta}^{(2)}}\right\} \quad \text{and} \quad C = -E\left\{\frac{\partial^2 \ell}{\partial \boldsymbol{\beta}^{(2)} \partial \boldsymbol{\beta}^{(2)'}}\right\}$$

both evaluated at $\hat{\boldsymbol{\beta}}^{(1)}$ and $\hat{\boldsymbol{\beta}}^{(2)}$. Often $\boldsymbol{\beta}^{(2)}$ has no more than two parameters and so no matrices larger than 2×2 need to be inverted in this calculation.

The definition of deviances given in Section 3.9 cannot be used when there is more than one systematic part since the maximum of the likelihood in the "saturated" model with all $\{\eta_i^{(1)}\}$ and $\{\eta_i^{(2)}\}$ allowed complete freedom is often infinite. For example, if $y_i \sim N(\eta_i^{(1)}, \eta_i^{(2)})$ then the saturated model estimates are $\hat{\eta}_i^{(1)} = y_i$ and $\hat{\eta}_i^{(2)} = 0$, resulting in infinite likelihood.

In models with a systematic part $\eta_i^{(1)}$ that can be factorized out, it is possible to test hypotheses about $\boldsymbol{\beta}^{(2)}$ without estimating $\boldsymbol{\beta}^{(1)}$ or using it in the test statistics. This can be done by defining the <u>proportional deviance</u> of a model to be

$$\text{proportional deviance} = 2(\ell_0^* - \ell^*(\hat{\boldsymbol{\beta}}^{(2)}))$$

where ℓ_0^* is the maximum of $\Sigma\, k_i(\eta_i^{(2)})$ over all possible $\eta_i^{(2)}$ and the functions $k_i(\cdot)$ are defined in (4.1.5); $\ell^*(\hat{\boldsymbol{\beta}}^{(2)})$ is its value at $\eta_i^{(2)}(\hat{\boldsymbol{\beta}}^{(2)})$, and $\hat{\boldsymbol{\beta}}^{(2)}$ is the maximum likelihood estimate for the model in question. With this definition, the difference in proportional deviances between nested models does not asymptotically have a χ^2 distribution as in Section 3.9, but it is asymptotically <u>proportional</u> to a χ^2 variate with degrees of freedom equal to the number of restricted parameters. An analysis of proportional deviance table formed as in Section 3.9 then has entries which are proportional to χ^2 variates and ratios of proportional deviances can be compared to F distributions in an identical way to standard analysis of variance. This was described by Nelder and Wedderburn(1972) and Jorgensen(1983). They used the term deviance instead of proportional deviance, but the latter term is given a more general definition below.

If one systematic part cannot be factorized out, proportional deviances cannot be defined as above. However we can define deviances to be the differences between $-2\log L(\beta^{(1)},\beta^{(2)})$ for the models in question and any convenient constant, and their differences can be interpreted as in Section 3.9, though the magnitudes of the deviances themselves are no longer meaningful. It should be noted that this definition of deviance in the presence of nuisance parameters is different from that of proportional deviances since our definition of deviance involves $\beta^{(1)}$ as well as $\beta^{(2)}$; differences in deviance are always directly compared to χ^2 critical values. Since $\beta^{(1)}$ and $\beta^{(2)}$ are treated more symmetrically with this definition, the deviances can be used for tests on $\beta^{(2)}$ as well as $\beta^{(1)}$.

4.3 EXAMPLES : NORMAL MODELS WITH VARIANCE A FUNCTION OF MEAN

One assumption of the standard normal linear model that is sometimes violated is the assumption of constant variance. A few approaches are possible such as transforming the response or maximizing quasi-likelihoods (Wedderburn, 1974). In this section we use a different approach and try to explicitly model the changes in variance. Our most general model is therefore one for n independent normal random variables $y_1,\ldots,y_n$ where y_i has mean $\mu_i(\beta)$ and variance $\sigma_i^2(\beta)$. The model therefore has log-likelihood components of the form,

$$\ell_i(\mu_i,\sigma_i^2) = -\{\log(\sigma_i^2) + \frac{(y_i-\mu_i)^2}{\sigma_i^2}\}/2 \ .$$

The model has two non-random systematic parts μ_i and σ_i^2, (or μ_i and any function of σ_i^2 such as $\log(\sigma_i^2)$ or σ_i^{-2}). Since

$$E\left[\frac{\partial^2\ell_i}{\partial\mu_i\,\partial\sigma_i^2}\right] = 0$$

the systematic parts are separable, the FS algorithm for β can be expressed as IRLS using (4.1.3). Since the FS algorithm with μ_i fixed is equivalent to the NRL algorithm with μ_i fixed and systematic part σ_i^{-2}, and the FS algorithm with σ_i fixed is equivalent to NRL with σ_i fixed and systematic part μ_i, it would also be expected that the asymptotic performence of the joint FS algorithm would depend on the linearity of the specifications for μ_i and σ_i^{-2} in the model being fitted.

We first consider models where the response variance is modelled as a function of the response mean, $\sigma_i^2 = v(\mu_i)$. Finney and Philips (1977) and Raab (1981) considered similar models in radioimmunoassay, but their methods were restricted to a single explanatory factor with replicates at each of its levels. We assume initially that $v(\cdot)$ is fully specified. Then the model can be written with a single systematic part and the methods of Chapter 3 can be used. For example,

if $\mu_i = \mathbf{x}_i'\boldsymbol{\beta}$ and $\sigma_i^2 = \mu_i^{\gamma}\sigma^2$ with γ and σ^2 known constants, then the FS algorithm can be implemented as IRLS with "explanatory" variables $\mathbf{x}_i$, and weights and "responses"

$$w_i = \sigma_i^{-2} + \gamma^2 / (2\mu_i^2) \tag{4.3.1}$$

and

$$z_i = \mu_i \left(\frac{\gamma(y_i-\mu_i)^2 + 2\mu_i(y_i-\mu_i) - \gamma\sigma_i^2}{2\mu_i^2 + \gamma^2\sigma_i^2} \right) .$$

In practice, however, unless the distribution of y_i is known to be for example Poisson, it is rarely possible to fully specify its variance in terms of its mean, so that auxiliary parameters must usually be incorporated in its form. The simplest generalization of this type of model is to specify $v(\cdot)$ up to a constant of proportionality, $v(\mu) = v_0(\mu)\sigma^2$. If the systematic parts are taken to be μ_i and σ^2, then σ^2 can be factorized out, reducing the problem to one with a single systematic part. However in most practical situations even this is more than can be specified with confidence. Often all that is known, or suspected, is that the response variance increases (or decreases) with its mean, in which case specifications such as

$$\sigma_i^2 = \alpha + \gamma\mu_i ,$$
$$\sigma_i^2 = (\alpha + \gamma\mu_i)^2$$

or

$$\sigma_i^2 = \mu_i^{\gamma}\sigma^2 = \exp\{\alpha + \gamma.\log \mu_i\} \tag{4.3.2}$$

may be more appropriate, where α and γ are unknown parameters. In each case constant variance corresponds to the parametric hypothesis that $\gamma=0$.

We now examine in detail the algorithms that can be used for maximum likelihood estimation in the model with $\mu_i = \mathbf{x}_i'\boldsymbol{\beta}^{(1)}$ and σ_i^2 given by (4.3.2). Writing $\boldsymbol{\beta}^{(2)\prime} = [\alpha : \gamma]$ and $\boldsymbol{\beta}' = [\boldsymbol{\beta}^{(1)\prime} : \boldsymbol{\beta}^{(2)\prime}]$, the FS iterations can be directly expressed as weighted least squares calculations using (4.1.3). These have a $2n\times p$ matrix of "explanatory" variables $\mathbf{X}$, a vector of $2n$ "responses" $\mathbf{z}$ and weights $w_1,\ldots,w_{2n}$, where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1' & 0 & 0 \\ \vdots & \vdots & \vdots \\ \mathbf{x}_n' & 0 & 0 \\ (\gamma/\mu_1)\mathbf{x}_1' & 1 & \log(\mu_1) \\ \vdots & \vdots & \vdots \\ (\gamma/\mu_n)\mathbf{x}_n' & 1 & \log(\mu_n) \end{bmatrix} \qquad \mathbf{z} = \begin{bmatrix} (y_1-\mu_1)\sigma_1^{-2}/w_1 \\ \vdots \\ (y_n-\mu_n)\sigma_n^{-2}/w_n \\ ((y_1-\mu_1)^2/\sigma_1^2-1)/(2w_{n+1}) \\ \vdots \\ ((y_n-\mu_n)^2/\sigma_n^2-1)/(2w_{2n}) \end{bmatrix}$$

and

$$\left.\begin{aligned} w_i &= \sigma_i^{-2} \\ w_{n+i} &= 0.5 \end{aligned}\right\} \quad \text{for } i=1,\ldots,n \ .$$

Since σ_i^{-1} is clearly nonlinear here, we next examine an NRL algorithm that uses a more linear function of σ_i^2 as the second systematic part. While still not linear, the systematic part $\log(\sigma_i^2)$ is closer to linearity than σ_i^{-2} and, if we ignore $\partial^2 \ell_i/\partial\mu_i\partial(\log\sigma_i^2)$ since the systematic parts are separable, the resulting NRL algorithm has "explanatory" variables $\mathbf{X}$ and "responses" $\mathbf{z}$ defined by the same equations as for FS, but with weights

$$\left.\begin{aligned} w_i &= \sigma_i^{-2} \\ w_{n+i} &= (y_i-\mu_i)^2\sigma_i^{-2}/2 \end{aligned}\right\} \quad \text{for } i=1,\ldots,n \ .$$

As with FS, this algorithm has positive weights since μ_i and $\log(\sigma_i^2)$ are both convex, and can be implemented with the least squares procedures in most existing statistical computer packages. The NRL algorithm of this form which does not treat $\partial^2 \ell_i/\partial\mu_i\partial(\log\sigma_i^2)$ as zero, adds an extra n rows of "data" to the least squares calculations with

$$\mathbf{X}^{(3)} = \begin{bmatrix} (\gamma/\mu_1+1)\mathbf{x}_1' & 1 & \log(\mu_1) \\ \vdots & \vdots & \vdots \\ (\gamma/\mu_n+1)\mathbf{x}_n' & 1 & \log(\mu_n) \end{bmatrix} \qquad \mathbf{z}^{(3)} = \mathbf{0}$$

and has weights defined by

$$\left.\begin{aligned} w_i &= \sigma_i^{-2}(\ 1 - (y_i-\mu_i)\ ) \\ w_{n+i} &= \sigma_i^{-2}(\ (y_i-\mu_i)^2/2 - (y_i-\mu_i)\ ) \\ w_{2n+i} &= \sigma_i^2(y_i-\mu_i) \end{aligned}\right\} \text{for } i=1,\ldots,n.$$

We shall next compare the performance of these three algorithms

(which will be denoted by FS, NRL-0 and NRL) and a fourth algorithm suggested by Stirling(1985), called FS-alternation, which alternates between the FS iterations (4.3.1) for $\beta^{(1)}$ with $\beta^{(2)}$ held fixed and the FS iterations for $\beta^{(2)}$ with $\beta^{(1)}$ held fixed, which are also simple least squares calculations. In each case, the starting values used were $\gamma = 0$ and the ordinary least squares estimates for the remaining parameters.

TABLE 4.3.1

Artificial Data for Fitting Normal Model with Variance = exp(α + γ.log(Mean))

Response, y_i	Explanatory Variable, d_i
3	1
1	2
6	3
2	4
9	5

The four algorithms were first applied to fit the model with $\mu_i = \beta_1+\beta_2 d_i$ to the artificial data in Table 4.3.1. Their iterations are shown in Tables 4.3.2 and 4.3.3. No switching between algorithms was allowed. To allow better comparison between the algorithms, stepsizes were only modified when a unit stepsize reduced the likelihood (i.e. the input parameter PACC to subroutine STEP was set as ρ=1.0). In this example, FS-alternation is intolerably slow to converge and FS and NRL-0 are not much better. The algorithm NRL is the only one that performs acceptably. If interpolation and extrapolation are allowed in the stepsize algorithm (ρ=0.1), all methods are somewhat improved. For example, Table 4.3.4 shows the iterations of the FS algorithm; the other algorithms are similarly

improved, but the relative performance of the algorithms is unaltered.

The algorithms were next applied to an example given by Ezekiel and Fox (1959) which related the stopping distance of cars to their speed. A plot of the data is given in Figure 4.3.1 showing both curvature and increasing variance. Ezekiel and Fox used the quadratic model $\mu_i = \beta_1 s_i + \beta_2 s_i^2$ to relate mean stopping distance to speed, s_i, and fitted this by unweighted least squares. Rutemiller and Bowers(1968) reanalysed the data assuming $\sigma_i^2 = \alpha + \gamma . s_i$. The analysis below assumes that response variance is proportional to a power of the response mean, a model that behaves better when expected stopping distance is small. Table 4.3.5 shows the iterations of FS and NRL (with ρ=1.0) for this example. As discussed in Section 3.5, all NRL algorithms, and also FS, have performance which approaches that of NR if there is a large amount of data and if the model fits the data well. In this example therefore, FS and NRL both converge acceptably well, though NRL is markedly better. Algorithm NRL-0 had an asymptotic convergence rate slightly better than that of FS, whereas FS-alternation performed slightly worse.

Table 4.3.5 also shows the log-likelihoods for the successive iterations. Twice the difference between the overall maximum (on convergence) and the maximum for γ=0 (iteration 0), 38.08, can be compared to the χ^2 distribution with one degree of freedom to test for heteroscedasticity and is highly significant.

Asymptotic formulae for variances and covariances of the estimates are given by $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ from the last iteration. From the FS algorithm, the estimated variances are

$$\text{var}\begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\alpha} \\ \hat{\gamma} \end{pmatrix} = \begin{bmatrix} 1.415\times10^{-2} & & & \\ -6.153\times10^{-4} & 3.418\times10^{-5} & & \\ -2.574\times10^{-2} & 1.102\times10^{-3} & 4.779\times10^{-1} & \\ 6.634\times10^{-3} & -3.144\times10^{-4} & -1.331\times10^{-1} & 3.990\times10^{-2} \end{bmatrix}$$

from which approximate confidence intervals can be found. The

TABLE 4.3.2

FS and NRL-O Algorithms Applied to Data in Table 4.3.1.
Starting Values (Iteration 0) were Found as Decribed in the Text.

Iteration	β_1	β_2	α	γ	Log-Likelihood	Convergence Rate, C
FS Algorithm						
0	0.30000	1.30000	1.64481	0.00000	-6.61201	
1	0.30000	1.30000	0.58572	0.80337	-6.35360	0.81036
2	1.03335	1.06094	-0.07414	1.18987	-6.23526	0.74634
3	1.06439	1.04100	-0.89272	1.74972	-6.14906	0.75935
4	1.33487	0.93198	-1.45964	2.14142	-6.11495	0.70332
5	1.39134	0.90292	-1.91741	2.47563	-6.09635	0.72337
6	1.49252	0.85608	-2.25071	2.72530	-6.08847	0.68144
7	1.52971	0.83660	-2.48397	2.90335	-6.08466	0.70072
8	1.56895	0.81702	-2.65206	3.03362	-6.08297	0.67401
9	1.58879	0.80651	-2.76553	3.12210	-6.08220	0.68545
10	1.60492	0.79814	-2.84484	3.18441	-6.08185	0.67185
:	:	:	:	:	:	:
21	1.63475	0.78226	-3.00691	3.31247	-6.08156	0.67011
22	1.63488	0.78219	-3.00759	3.31301	-6.08156	0.67016
23	1.63496	0.78215	-3.00804	3.31337	-6.08156	0.67033
24	1.63501	0.78212	-3.00834	3.31361	-6.08156	0.67056
25	1.63505	0.78210	-3.00854	3.31377	-6.08156	0.67093
NRL-O Algorithm						
0	0.30000	1.30000	1.64481	0.00000	-6.61201	
1	0.30000	1.30000	-0.37385	1.32110	-6.39675	0.65994
2	1.40936	0.91590	-0.71792	1.61660	-6.20111	0.67089
3	1.11151	1.01765	-2.06626	2.53117	-6.16597	0.57147
4	1.66671	0.77944	-2.23924	2.72835	-6.11187	0.55926
5	1.41922	0.88403	-2.68205	3.02977	-6.09996	0.66858
6	1.68230	0.76379	-2.73885	3.11387	-6.08979	0.57665
7	1.53733	0.82873	-2.89458	3.20926	-6.08616	0.67881
8	1.66813	0.76794	-2.90929	3.24335	-6.08378	0.59813
9	1.58896	0.80408	-2.96933	3.27535	-6.08275	0.67375
10	1.65490	0.77330	-2.97079	3.28845	-6.08215	0.63000
:	:	:	:	:	:	:
24	1.63537	0.78194	-3.00885	3.31406	-6.08156	0.81445
25	1.63494	0.78214	-3.00898	3.31408	-6.08156	0.63009
26	1.63525	0.78200	-3.00891	3.31408	-6.08156	0.79967
27	1.63503	0.78210	-3.00897	3.31409	-6.08156	0.64232
28	1.63519	0.78203	-3.00893	3.31409	-6.08156	0.79448

corresponding estimates from the NRL iterations are

$$\text{var}\begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\alpha} \\ \hat{\gamma} \end{pmatrix} = \begin{bmatrix} 1.432\times10^{-2} & & & \\ -6.245\times10^{-4} & 3.467\times10^{-5} & & \\ -3.576\times10^{-2} & 1.591\times10^{-3} & 5.415\times10^{-1} & \\ 9.664\times10^{-3} & -4.622\times10^{-4} & -1.517\times10^{-1} & 4.537\times10^{-2} \end{bmatrix}$$

In these examples, NRL based on systematic parts μ_i and $\log(\sigma_i^2)$ is better than the FS algorithm. However, since ($\mathbf{X}'\mathbf{W}\mathbf{X}$) may not be positive definite for the former algorithm, but is always positive definite for the latter, FS may perform better in the initial iterations for some examples. The safest algorithm would allow switching between these two algorithms, depending on the rate of convergence, as described in Section 3.7.

TABLE 4.3.3

NRL and FS-alternation Algorithms Applied to Data in Table 4.3.1. Starting Values (Iteration 0) were Found as Decribed in the Text. (Bracketed Iterations had Stepsize Modifications Applied)

Iteration	β_1	β_2	α	γ	Log-Likelihood	Convergence Rate, C
NRL Algorithm						
0	0.30000	1.30000	1.64481	0.00000	-6.61201	
1	(3.20011	0.43030	-2.54102	2.73939	-7.00120)	
	1.44090	0.95786	-0.00189	1.07767	-6.33729	0.57159
2	1.40280	0.84981	-2.87854	3.03501	-6.22919	0.12641
3	1.60335	0.78517	-3.09148	3.35962	-6.08550	0.22963
4	1.62795	0.78474	-2.99740	3.30423	-6.08158	0.19196
5	1.63526	0.78194	-3.01075	3.31553	-6.08156	0.11148
6	1.63505	0.78210	-3.00861	3.31382	-6.08156	0.21102
7	1.63514	0.78205	-3.00901	3.31414	-6.08156	0.17077
8	1.63512	0.78206	-3.00894	3.31408	-6.08156	0.22570
FS-alternation Algorithm						
0	0.30000	1.30000	1.64481	0.00000	-6.61201	
1			0.58572	0.80337	-6.35360	
	0.73799	1.13419			-6.33226	0.74888
2			-0.23376	1.32448	-6.21766	
	1.00020	1.04386			-6.20943	0.76979
3			-0.89569	1.76707	-6.15285	
	1.18677	0.97666			-6.14765	0.76008
4			-1.41180	2.12269	-6.11870	
	1.30887	0.93059			-6.11597	0.75834
5			-1.79116	2.39034	-6.10186	
	1.39247	0.89750			-6.10034	0.76597
6			-2.06915	2.59081	-6.09317	
	1.45065	0.87321			-6.09230	0.77560
7			-2.27385	2.74141	-6.08850	
	1.49229	0.85490			-6.08798	0.78582
8			-2.42679	2.85598	-6.08586	
	1.52281	0.84083			-6.08554	0.79513
9			-2.54294	2.94435	-6.08431	
	1.54569	0.82983			-6.08410	0.80320
10			-2.63254	3.01345	-6.08335	
	1.56318	0.82113			-6.08321	0.80996
⋮	⋮	⋮	⋮	⋮	⋮	⋮
27			-2.99242	3.30052	-6.08156	
	1.63202	0.78385			-6.08156	0.83950
28			-2.99507	3.30270	-6.08156	
	1.63252	0.78357			-6.08156	0.83970
29			-2.99730	3.30452	-6.08156	
	1.63294	0.78333			-6.08156	0.83988
⋮	⋮	⋮	⋮	⋮	⋮	⋮

TABLE 4.3.4

FS Algorithm with Stepsize improvement Applied to Data in Table 4.3.1. Starting Values (Iteration 0) were Found as Decribed in the Text. (Bracketed Iterations had Stepsize Modifications Applied)

Iteration	β_1	β_2	α	γ	Log-Likelihood	Convergence Rate, C
0	0.30000	1.30000	1.64481	0.00000	-6.61201	
1 (	0.30000	1.30000	0.58572	0.80337	-6.35360)	
	0.30000	1.30000	0.27466	1.03933	-6.34148	0.75466
2 (	1.21294	0.99430	-0.31993	1.35535	-6.21992)	
	1.10872	1.02919	-0.25206	1.31928	-6.21605	0.74550
3 (	1.10903	1.02301	-1.04394	1.85532	-6.13862)	
	1.10921	1.01937	-1.50991	2.17075	-6.13054	0.61637
4 (	1.50593	0.85568	-1.89846	2.46615	-6.10378)	
	1.41144	0.89467	-1.80592	2.39579	-6.10059	0.72164
5 (	1.45334	0.87369	-2.16960	2.66409	-6.08993)	
	1.49655	0.85206	-2.54457	2.94071	-6.08549	0.42575
6 (	1.59990	0.80273	-2.70735	3.07786	-6.08300)	
	1.58232	0.81112	-2.67966	3.05454	-6.08288	0.64102
7 (	1.58787	0.80667	-2.77825	3.13191	-6.08213)	
	1.59250	0.80295	-2.86039	3.19638	-6.08192	0.49170
8 (	1.62532	0.78795	-2.92147	3.24525	-6.08168)	
	1.61996	0.79040	-2.91150	3.23727	-6.08167	0.59975
9 (	1.62152	0.78920	-2.94115	3.26030	-6.08160)	
	1.62284	0.78820	-2.96604	3.27963	-6.08159	0.48439
10 (	1.63237	0.78373	-2.98403	3.29445	-6.08157)	
	1.63083	0.78445	-2.98114	3.29207	-6.08157	0.59004
11 (	1.63125	0.78410	-2.98954	3.29867	-6.08156)	
	1.63161	0.78380	-2.99662	3.30424	-6.08156	0.48552
12 (	1.63435	0.78253	-3.00190	3.30853	-6.08156)	
	1.63391	0.78273	-3.00105	3.30784	-6.08156	0.58458
13 (	1.63403	0.78263	-3.00345	3.30972	-6.08156)	
	1.63413	0.78255	-3.00547	3.31130	-6.08156	0.48394
14 (	1.63491	0.78219	-3.00696	3.31252	-6.08156)	
	1.63478	0.78225	-3.00672	3.31233	-6.08156	0.58429
15 (	1.63482	0.78222	-3.00740	3.31286	-6.08156)	
	1.63484	0.78220	-3.00797	3.31330	-6.08156	0.48476
16 (	1.63506	0.78209	-3.00839	3.31365	-6.08156)	
	1.63503	0.78211	-3.00832	3.31359	-6.08156	0.58421
17 (	1.63504	0.78210	-3.00851	3.31374	-6.08156)	
	1.63505	0.78210	-3.00867	3.31387	-6.08156	0.48570
18 (	1.63511	0.78207	-3.00879	3.31397	-6.08156)	
	1.63510	0.78207	-3.00877	3.31395	-6.08156	0.58638

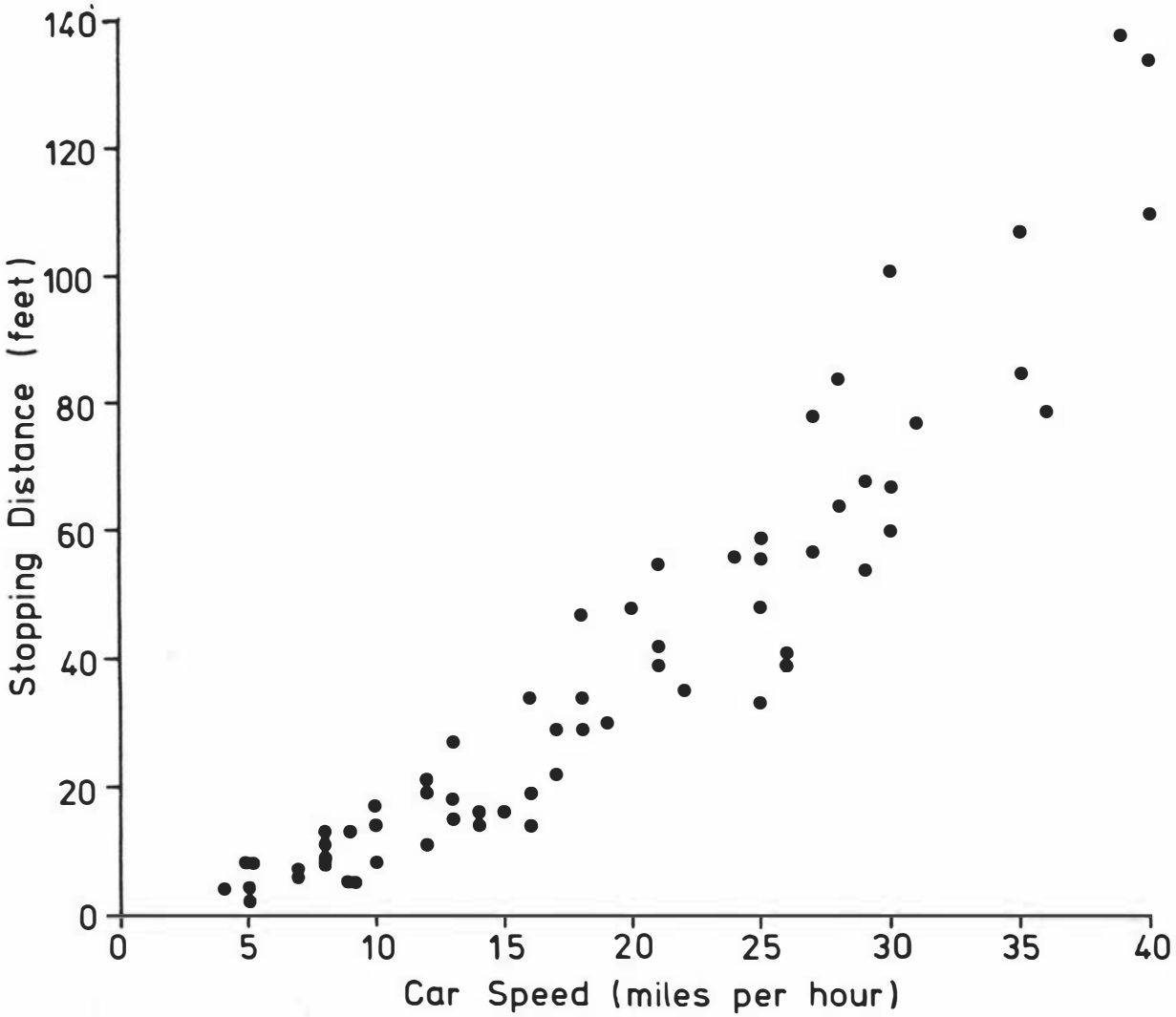

FIGURE 4.3.1 Data for Stopping Distance Example

TABLE 4.3.5

FS and NRL Algorithms Applied to Stopping Distance Example

Starting Values (Iteration 0) were Found as Decribed in the Text.

(Bracketed Iterations had Stepsize Modifications Applied)

Iteration	β_1	β_2	α	γ	Log-Likelihood	Convergence Rate, C
FS Algorithm						
0	0.55526	0.06269	4.53674	0.00000	-174.40727	
1	0.55526	0.06269	1.76048	0.84734	-161.91077	0.45397
2	0.61394	0.06050	0.50868	1.10415	-156.73634	0.47904
3	0.62912	0.05985	-0.30610	1.29953	-155.47225	0.28909
4	0.64226	0.05925	-0.57292	1.36721	-155.37461	0.16531
5	0.64676	0.05903	-0.61541	1.37883	-155.37218	0.14704
6	0.64745	0.05900	-0.62153	1.38062	-155.37212	0.14876
7	0.64756	0.05899	-0.62243	1.38088	-155.37212	0.15020
8	0.64758	0.05899	-0.62257	1.38093	-155.37212	0.14983
9	0.64758	0.05899	-0.62259	1.38093	-155.37212	0.15017
NRL Algorithm						
0	0.55526	0.06269	4.53674	0.00000	-174.40727	
1 (	0.73634	0.05644	-4.85355	2.31495	-219.39314)	
	0.60581	0.06095	1.91523	0.64627	-161.43867	0.49971
2	0.64551	0.05927	-1.24475	1.50083	-156.47864	0.22447
3	0.64692	0.05904	-0.73873	1.40759	-155.39465	0.19277
4	0.64751	0.05900	-0.62612	1.38180	-155.37214	0.03115
5	0.64758	0.05899	-0.62259	1.38093	-155.37212	0.00061
6	0.64758	0.05899	-0.62259	1.38093	-155.37212	0.00157

4.4 EXAMPLES : NORMAL MODELS WITH VARIANCE A FUNCTION OF EXPLANATORY VARIABLES

We next consider a different kind of model for heteroscedasticity in normal models, with $\mu_i = \mu_i(\boldsymbol{\beta}^{(1)})$ and $\sigma_i^2 = \sigma_i^2(\boldsymbol{\beta}^{(2)})$, and we restrict attention to models with linear systematic parts,

$$\begin{aligned} \mu_i &= m(\mathbf{x}_i^{(1)}, \boldsymbol{\beta}^{(1)}) \\ \sigma_i^2 &= v(\mathbf{x}_i^{(2)}, \boldsymbol{\beta}^{(2)}) \end{aligned} \tag{4.4.1}$$

where the explanatory variables in the vectors $\mathbf{x}_i^{(2)}$ may possibly be related to those in $\mathbf{x}_i^{(1)}$. Models of this form have been considered by Rutemiller and Bowers(1968) and Harvey(1976) and include random coefficients models (Hildreth and Houck, 1968) and models where the measured response y_i is a total over a group of n_i related units with between group variance σ_b^2 and within group variance σ_w^2, so that $\sigma_i^2 = n_i\sigma_w^2 + n_i^2\sigma_b^2$.

This model has two linear systematic parts that are detached. Whether of not the systematic parts are convex depends on the functions $m(\cdot)$ and $v(\cdot)$. For example $(\mathbf{x}_i^{(1)}, \boldsymbol{\beta}^{(1)})$ is convex if $m(\eta) = \eta$ and $(\mathbf{x}_i^{(2)}, \boldsymbol{\beta}^{(2)})$ is convex if $v(\eta) = \exp(\eta)$ or $v(\eta) = \eta^{-1}$.

As in the models in Section 4.3, the asymptotic performance of the FS algorithm is expected to depend on the linearity of μ_i and σ_i^{-2}. The NRL algorithm applied to systematic parts $\mathbf{x}_i^{(1)}, \boldsymbol{\beta}^{(1)}$ and $\mathbf{x}_i^{(2)}, \boldsymbol{\beta}^{(2)}$ is equivalent to NR and has quadratic convergence. Since the two systematic parts are detached, FS can be slightly improved on by FS-alternation which alternates between FS iterations for $\boldsymbol{\beta}^{(1)}$ with $\boldsymbol{\beta}^{(2)}$ held fixed and FS iterations for $\boldsymbol{\beta}^{(2)}$ with $\boldsymbol{\beta}^{(1)}$ held fixed. When $\mu_i = \mathbf{x}_i^{(1)}, \boldsymbol{\beta}^{(1)}$ and $\sigma_i^2 = \mathbf{x}_i^{(2)}, \boldsymbol{\beta}^{(2)}$, both FS and NR iterations for $\boldsymbol{\beta}^{(1)}$ with $\boldsymbol{\beta}^{(2)}$ held fixed are identical and are equivalent to the explicit maximum likelihood solutions found by least squares with responses $\{y_i\}$, explanatory variables $\{\mathbf{x}_i^{(1)}\}$ and weights $\{ (\mathbf{x}_i^{(2)}, \boldsymbol{\beta}^{(2)})^{-1} \}$. The

FS iterations for $\boldsymbol{\beta}^{(2)}$ with $\boldsymbol{\beta}^{(1)}$ fixed have "explanatory" variables $\mathbf{x}_i^{(2)}$, "responses"

$$z_i^{(2)} = \frac{(y_i-\mu_i)^2 - \sigma_i^2}{2\,\sigma_i^2\, w_i^{(2)}}$$

and weights

$$w_i^{(2)} = \frac{1}{2\,\sigma_i^4}$$

The joint NR algorithm has $3n \times p$ matrix of "explanatory variables $\mathbf{X}$, vector of $3n$ "responses" $\mathbf{z}$ and weights $w_1,\ldots,w_{3n}$ where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^{(1)}, & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{x}_n^{(1)}, & \mathbf{0} \\ \mathbf{0} & \mathbf{x}_1^{(2)}, \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{x}_n^{(2)}, \\ \mathbf{x}_1^{(1)}, & \mathbf{x}_1^{(2)}, \\ \vdots & \vdots \\ \mathbf{x}_n^{(1)}, & \mathbf{x}_n^{(2)}, \end{bmatrix} \qquad \mathbf{z} = \begin{bmatrix} (y_1-\mu_1)/(\sigma_1^2 w_1) \\ \vdots \\ (y_n-\mu_n)/(\sigma_n^2 w_n) \\ ((y_1-\mu_1)^2-\sigma_1^2)/(2\sigma_1^4 w_{n+1}) \\ \vdots \\ ((y_n-\mu_n)^2-\sigma_n^2)/(2\sigma_n^4 w_{2n}) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

and

$$\left.\begin{aligned} w_i &= (\sigma_i^2 - (y_i-\mu_i)) \,/\, \sigma_i^4 \\ w_{n+i} &= ((y_i-\mu_i)^2 - (y_i-\mu_i-0.5)\sigma_i^2) \,/\, \sigma_i^6 \\ w_{2n+i} &= (y_i-\mu_i) \,/\, \sigma_i^4 \end{aligned}\right\} \quad \text{for } i=1,\ldots,n$$

Starting values can be found by noting that (4.1.6) depends on the value of $\boldsymbol{\beta}$ in the previous iterations only through the two systematic parts. $\boldsymbol{\beta}^{(1)}$ can be started from its unweighted least squares estimate and $(\mathbf{x}_i^{(2)}\boldsymbol{\beta}^{(2)})$ can be replaced by its maximum likelihood estimate from the saturated model with fixed $\boldsymbol{\beta}^{(1)}$, $(y_i-\mathbf{x}_i^{(1)}\boldsymbol{\beta}^{(1)})^2$.

The maximum likelihood estimators are asymptotically independent and $\mathrm{var}(\hat{\boldsymbol{\beta}}^{(1)})$ and $\mathrm{var}(\hat{\boldsymbol{\beta}}^{(1)})$ can be consistently estimated by the inverse matrices used in the last FS iterations for $\boldsymbol{\beta}^{(1)}$ with $\boldsymbol{\beta}^{(2)}$ fixed and for $\boldsymbol{\beta}^{(2)}$ with $\boldsymbol{\beta}^{(1)}$ fixed, respectively. Alternatively, $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ from the final joint NRL iteration can be used.

We next consider the details of applying the algorithm to normal

models with random block effects, which will be shown to be closely related to heteroscedastic models. We deal here with observations that satisfy the model

$$y_{ij} = \mathbf{x}_{ij}'\boldsymbol{\beta} + b_i + e_{ij} \quad i=1,\ldots,a;\ j=1,\ldots,r_i \qquad (4.4.2)$$

where $e_{ij} \sim N(0,\phi_e)$, $b_i \sim N(0,\phi_b)$, all $\{e_{ij}\}$ and $\{b_i\}$ are independent, $\boldsymbol{\beta}$ is a vector of unknown parameters and ϕ_e and ϕ_b are unknowm scalar parameters. In an obvious matrix notation the model can be written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \mathbf{e}$$

where $\mathbf{b}$ is the vector of random effects.

Harville(1977) describes several algorithms for maximum likelihood estimation of parameters in random effects models, the faster ones being based on expressions he gives for the partial derivatives of the log-likelihood. We next show how model (4.4.2) can be re-expressed as a heteroscedastic model to allow use of the IRLS algorithms that were described at the beginning of this section. The algorithms, being based on FS or NR, are competitive with the others described by Harville in iterations to convergence, and are relatively easy to implement. Also estimates of variances and covariances of the parameter estimates are immediately available.

An orthogonal transformation can be applied to $[\mathbf{Z} : \mathbf{X} : \mathbf{y}]$ with (n-a) Givens rotations such that

$$\mathbf{Q}_1 [\ \mathbf{Z} : \mathbf{X} : \mathbf{y}\ ] = \begin{bmatrix} \mathbf{D} \\ \mathbf{0} \end{bmatrix} : \mathbf{X}^* : \mathbf{y}^*$$

where $\mathbf{D}$ is an $a \times a$ diagonal matrix with diagonal elements $\sqrt{r_1},\ldots,\sqrt{r_a}$. The conditional distribution of $\mathbf{y}^*$ given $\mathbf{b}$ is therefore normal with mean

$$E[\mathbf{y}^* \mid \mathbf{b}] = \begin{bmatrix} \sqrt{r_1}b_1 \\ \sqrt{r_2}b_2 \\ \vdots \\ \sqrt{r_a}b_a \\ \mathbf{0} \end{bmatrix} + \mathbf{X}^*\boldsymbol{\beta} \qquad (4.4.3)$$

and variance $\mathbf{I}\phi_e$. Therefore unconditionally $\mathbf{y}^*$ is a vector of independent normal observations with mean $\mathbf{X}^*\boldsymbol{\beta}$ and variances $(\phi_e+r_i\phi_b)$ for $i=1,\ldots,n$ where $r_{a+1},\ldots,r_n$ are defined for convenience to be zero. In terms of $\mathbf{y}^*$ the model is therefore heteroscedastic and the algorithms above can be applied.

Matrices $\mathbf{Q}_1$, $\mathbf{Z}$ and $\mathbf{D}$ are not explicitly required. $\mathbf{X}^*$ and $\mathbf{y}^*$ can be generated by the following algorithm:

$r_i := 0 \qquad (i=1,\ldots,a)$
$k := a$
for each response y_{ij}, presented in any order do
  $r_i := r_i+1$
  if $r_i=1$ then
    $[\mathbf{x}_i^{*\prime} : y_i^*] := [\mathbf{x}_{ij}' : y_{ij}]$
  else
    $k := k+1$
    $s := \sqrt{r_i}$; $c := \sqrt{(1-1/r_i)}$
    $[\mathbf{x}_k^{*\prime} : y_k^*] := c[\mathbf{x}_{ij}' : y_{ij}] - s[\mathbf{x}_i^{*\prime} : y_i^*]$
    $[\mathbf{x}_i^{*\prime} : y_i^*] := c[\mathbf{x}_i^{*\prime} : y_i^*] + s[\mathbf{x}_{ij}' : y_{ij}]$ (4.4.4)
  end if
end for

The basic algorithm can be made more efficient in several ways.

(1) $\mathbf{X}$ is often sparse; storage is only required for a single row if $\mathbf{x}_{ij}$ is generated inside the loop. Alternatively the algorithm can be easily modified to overwrite $\mathbf{X}$ and $\mathbf{y}$ with $\mathbf{X}^*$ and $\mathbf{y}^*$.

(2) Execution time during the iterative part of the algorithm, and storage, can be saved if some further preliminary reduction to the data is done. Orthogonal transformations can be applied to

the last (n-a) rows of $[\mathbf{X}^* : \mathbf{y}^*]$ such that

$$\mathbf{Q}_2 [\mathbf{X}^* : \mathbf{y}^*] = \begin{bmatrix} \mathbf{X}^0 & \mathbf{y}^0 \\ 0 & S \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where $\mathbf{X}^0$ is a $(p+a)\times p$ matrix with its last p rows upper triangular, $\mathbf{y}^0$ is a vector of (p+a) elements, S is a scalar and $\mathbf{Q}_2$ is an $n\times n$ orthogonal matrix. The most efficient way of doing this is with Givens rotation on the rows $[\mathbf{x}_k^{*\prime} : y_k^*]$ as soon as they are formed in (4.4.4); the rows do not then need to be stored. If GIVEN3 is used to form the last p rows of $[\mathbf{X}^0 : \mathbf{y}^0]$ from them, only $(p+1)(a+p/2+1)$ storage locations are needed.

Then $\mathbf{y}^0 \sim N(\mathbf{X}^0\boldsymbol{\beta}, \text{diag}(\phi_e + r_i\phi_b))$ and $S^2 \sim \phi_e\chi^2(n-a-b)$ are (p+a+1) sufficient statistics. The iterations for $\boldsymbol{\beta}$ with ϕ_e and ϕ_b fixed can be based on fitting a normal distribution to $\mathbf{y}^0$ whereas the iterations for ϕ_e and ϕ_b with $\boldsymbol{\beta}$ fixed are based on fitting gamma distributions to the (p+a) squared residuals and S^2.

(3) The Givens rotations in (4.4.4) and (2) above can be replaced by the faster versions of the Givens rotations described in Section 2.3; these are competitive in speed with methods based on Householder transformations.

To illustrate the method, it was applied to an artificial data set given by Cunningham and Henderson(1968) and shown in Table 4.4.1. Table 4.4.2 shows the matrices $\mathbf{Z}^*$ and $\mathbf{y}^*$ and the values $r_1,\ldots,r_n$ for the random effects model with mean μ_i for treatment i (i=1,2). Table 4.4.3 shows the FS-alternation and NR iterations applied to Table 4.4.2. Both algorithms used interolation or extrapolation to get improved step sizes (ρ=0.1) in each iteration. Starting values were found as described at the beginning of this section. The NR algorithm has relatively poor initial convergence. Since FS also has a good asymptotic convergence rate in this example, it would be preferred. However, switching between the two algorithms would avoid NR's poor

TABLE 4.4.1

Artificial data from Cunningham and Henderson(1968)

Block	Treatment	
	1	2
1	3,2	2,3
2	2,3,5,6,7	8,8,9
3	3	4,4,3,2,5

initial convergence and would give a safeguard against the chance of poor asymptotic convergence with FS.

The variance-covariance matrix of the estimates is estimated from the FS algorithm as

$$\text{var}\begin{Bmatrix}\hat{\mu}_1\\ \hat{\mu}_2\\ \hat{\phi}_e\\ \hat{\phi}_b\end{Bmatrix} = \begin{bmatrix}1.174 & 0.822 & 0 & 0\\ 0.822 & 1.089 & 0 & 0\\ 0 & 0 & 0.737 & -0.130\\ 0 & 0 & -0.130 & 5.709\end{bmatrix}$$

The corresponding estimate from the NR algorithm is

$$\text{var}\begin{Bmatrix}\hat{\mu}_1\\ \hat{\mu}_2\\ \hat{\phi}_e\\ \hat{\phi}_b\end{Bmatrix} = \begin{bmatrix}1.190 & 0.822 & 0.043 & -0.276\\ 0.822 & 1.089 & -0.002 & 0.010\\ 0.043 & -0.002 & 0.736 & -0.120\\ -0.276 & 0.010 & -0.120 & 5.642\end{bmatrix}$$

The log-likelihood for this model is -36.1945 (14 d.f.) whereas those for the two models with constraints $\mu_1=\mu_2$ and $\phi_b=0$ are -38.4579 (15 d.f.) and -39.6662 (15 d.f.) respectively. Twice the differences between these and the unconstrained model's deviance are 4.527 and 6.943, both with 1 d.f. so that treatment and block effects are significant at 5 and 1 per cent significance levels respectively.

TABLE 4.4.2

Data in Table 4.4.1 Transformed to independence

y^*	x^*		r_i
-0.707106781	0.000000000	0.000000000	0
-0.408248290	-0.816496581	0.816496581	0
0.577350269	-0.577350269	0.577350269	0
0.707106781	0.000000000	0.000000000	0
2.041241452	0.000000000	0.000000000	0
2.309401077	0.000000000	0.000000000	0
2.683281573	0.000000000	0.000000000	0
3.103761159	-0.912870929	0.912870929	0
2.623156949	-0.771516750	0.771516750	0
3.207134903	-0.668153105	0.668153105	0
0.707106781	-0.707106781	0.707106781	0
0.408248290	-0.408248290	0.408248290	0
-0.577350269	-0.288675135	0.288675135	0
-1.341640786	-0.223606798	0.223606798	0
1.643167673	-0.182574186	0.182574186	0
5.000000000	1.000000000	1.000000000	4
16.970562748	1.767766953	1.060660172	8
8.573214100	0.408248290	2.041241452	6

A related method was described by Patterson and Thompson (1971), who showed that it is possible to find a $(n-p)\times n$ matrix $\mathbf{P}$ such that the elements of $\mathbf{Py}$ are independent with zero means and variances that are linear in ϕ_e and ϕ_b. They proposed estimating ϕ_e and ϕ_b to maximize the likelihood of these $(n-p)$ error contrasts; the method, called restricted maximum likelihood, is unique since all such $\mathbf{Py}$ with different $\mathbf{P}$ are linear transformations of each other. There has been recently some discussion about whether this should be preferred to full maximum likelihood (e.g. Harville (1977) and the discussion following it and Swallow and Modahan (1984)). A suitable $\mathbf{Py}$ cannot however be easily found in practice and restricted maximum likelihood algorithms are usually based on other computational approaches.

TABLE 4.4.3

FS-alternation and NR Algorithms Applied to Data in Table 4.4.2
Starting Values (Iteration 0) were Found as Decribed in the Text.
(Bracketed Iterations had Stepsize Modifications Applied)

Iteration	μ_1	μ_2	ϕ_e	ϕ_b	Log-Likelihood	Convergence Rate, C
FS-alternation						
0	3.87500	4.80000	0.00127	1.78929	-15648.2770	
1	2.84843	4.87078			-13887.3334	0.55193
			2.35543	2.45631	-19.67247	
			(1.17898	2.12298	-21.99355)	
2	2.98893	4.86652			-19.65379	0.01646
			2.35198	2.50378	-19.65359	
3	2.98645	4.86661			-19.65359	0.01349
			2.35207	2.50319	-19.65359	
4	2.98648	4.86661			-19.65359	0.01343
			2.35207	2.50320	-19.65359	
5	2.98648	4.86661			-19.65359	0.01304
NR Algorithm						
0	3.20428	4.20370	0.00127	1.78929	-15417.44361	
1 (	3.04757	4.61476	0.00183	2.26339	-9809.71043)	
	2.97784	4.79767	0.00208	2.47434	-8475.65254	0.62260
2 (	2.90753	4.82904	0.00312	2.76264	-5630.59200)	
	2.87254	4.84465	0.00364	2.90613	-4823.48126	1.17539
3 (	2.86296	4.86005	0.00545	2.86871	-3207.03372)	
	2.85817	4.86775	0.00636	2.85002	-2745.69827	0.97739
4 (	2.85246	4.86811	0.00953	2.87685	-1823.58613)	
	2.84960	4.86829	0.01112	2.89026	-1560.44861	1.01580
5 (	2.85019	4.87010	0.01667	2.87813	-1034.82787)	
	2.85048	4.87100	0.01944	2.87207	-884.96659	0.99139
6 (	2.85025	4.87058	0.02912	2.87671	-585.90054)	
	2.85014	4.87038	0.03394	2.87903	-500.76812	0.99724
7 (	2.85136	4.87068	0.05079	2.87341	-331.15771)	
	2.85197	4.87083	0.05918	2.87061	-283.01118	0.98764
8 (	2.85354	4.87066	0.08840	2.86746	-187.36868)	
	2.85432	4.87058	0.10289	2.86590	-160.35333	0.98177
9 (	2.85734	4.87058	0.15319	2.85715	-106.96632)	
	2.85884	4.87058	0.17800	2.85284	-92.01957	0.96627
10 (	2.86384	4.87043	0.26350	2.83958	-62.75754)	
	2.86628	4.87036	0.30522	2.83311	-54.69581	0.94177
11 (	2.87466	4.87017	0.44726	2.81026	-39.18205)	
	2.87866	4.87008	0.51519	2.79933	-35.03443	0.89746
12 (	2.89194	4.86972	0.74114	2.76327	-27.30945)	
	2.89806	4.86956	0.84512	2.74667	-25.36200	0.82066
13 (	2.91745	4.86900	1.17519	2.69363	-21.96544)	
	2.92569	4.86876	1.31537	2.67110	-21.20933	0.68822
14 (	2.94930	4.86799	1.71784	2.60615	-20.06687)	
	2.95762	4.86771	1.85972	2.58325	-19.87819	0.47523
15 (	2.97649	4.86701	2.18167	2.53096	-19.67539)	
	2.98056	4.86686	2.25098	2.51970	-19.66095	0.20546
16	2.98599	4.86663	2.34374	2.50456	-19.65363	0.08240
17	2.98648	4.86661	2.35201	2.50321	-19.65359	0.00705
18	2.98648	4.86661	2.35207	2.50320	-19.65359	0.00000

4.5 EXAMPLES : NEGATIVE BINOMIAL MODELS

In this section we consider parameter estimation for models in which $y_1,\ldots,y_n$ are independent negative binomial variables with means $E[y_i] = \mu_i = \mu(\eta_i^{(1)})$ and variances $\text{var}(y_i) = \mu_i(1+\eta_i^{(2)})$ where $\eta_i^{(1)} = \mathbf{x}_i'\boldsymbol{\beta}$ and $\eta_i^{(2)} = (\phi_1+\phi_2\mu_i)$ for $i=1,\ldots,n$. The probability function of y_i is therefore

$$p(y_i \mid \eta_i^{(1)},\eta_i^{(2)}) = \binom{y_i+\mu_i/\eta_i^{(2)}-1}{y_i} (1+\eta_i^{(2)})^{-\mu_i/\eta_i^{(2)}} \left(\frac{\eta_i^{(2)}}{1+\eta_i^{(2)}}\right)^{y_i}$$

for $y_i = 0,1,\ldots.$ and its log-likelihood components can be written as

$$\ell_i(\eta_i^{(1)},\eta_i^{(2)}) = \psi\left(y_i+\frac{\mu_i}{\eta_i^{(2)}}\right)-\psi\left(\frac{\mu_i}{\eta_i^{(2)}}\right)-\frac{\mu_i}{\eta_i^{(2)}}\log(1+\eta_i^{(2)})+y_i\log\left(\frac{\eta_i^{(2)}}{1+\eta_i^{(2)}}\right)$$

where $\psi(\cdot) = \log\ \Gamma(\cdot)$. It is well known that the negative binomial distribution approaches the Poisson distribution as its variance becomes near its mean ($\eta_i^{(2)} = 0$). To deal with $\eta_i^{(2)}$ near zero stably, it is convenient to reexpress the log-likelihood components as

$$\ell_i(\eta_i^{(1)},\eta_i^{(2)}) = A\left(\frac{\eta_i^{(2)}}{\mu_i},y_i\right) - \mu_i B(\eta_i^{(2)}) + y_i\log\left(\frac{\mu_i}{1+\eta_i^{(2)}}\right) \qquad (4.5.1)$$

where

$$A(a,b) = \sum_{j=0}^{b-1} \log\,(1+ja)$$

and

$$B(a) = \begin{cases} \dfrac{\log(1+a)}{a} & \text{if } a\neq 0 \\ 1 & \text{if } a=0 \end{cases}.$$

The two special cases of this model where $\phi_1 = 0$ and where $\phi_2 = 0$ correspond to the models (d)(1) and (d)(2) respectively in Section 3.2. The hypothesis of $\phi_1 = 0$ can therefore be treated as a test for whether the counts y_i are composed of clusters and the hypothesis of $\phi_2 = 0$ tests whether there are systematic changes between the experimental

units that are not explained by the proposed function $\mu_i(\beta)$. If both $\phi_1 = 0$ and $\phi_2 = 0$ then the distribution is Poisson.

If $\phi_1 = 0$ and ϕ_2 is known, the distribution is in Nelder and Wedderburn's (1972) exponential family of Generalized Linear Models and FS is equivalent to NRL based on the single systematic part $\log(\phi_2+\mu_i^{-1})$. However in all other cases, the expected second derivatives cannot be easily found and FS is therefore not practical. We therefore only consider NRL algorithms for this problem.

If ϕ_1 or ϕ_2 are fixed, then the model can be expressed with two linear systematic parts, $\eta_i^{(1)}$ and the other unknown ϕ_i, and NRL applied to these is equivalent to NR. If both ϕ_1 and ϕ_2 must be estimated, then NR is equivalent to NRL based on the three systematic parts $\eta_i^{(1)}$, ϕ_1 and ϕ_2. Since it is more difficult to apply NRL based on three systematic parts, we shall instead describe the NRL algorithm based on the two systematic parts $\eta_i^{(1)}$ and $\eta_i^{(2)}$. This algorithm can be expressed as IRLS using (4.1.1). It depends on the derivatives

$$\frac{\partial \ell_i}{\partial \eta_i^{(1)}} = \left\{ -A'\left(\frac{\eta_i^{(2)}}{\mu_i}, y_i\right).\frac{\eta_i^{(2)}}{\mu_i^2} - B(\eta_i^{(2)}) + \frac{y_i}{\mu_i} \right\} \mu_i'$$

$$\frac{\partial \ell_i}{\partial \eta_i^{(2)}} = A'\left(\frac{\eta_i^{(2)}}{\mu_i}, y_i\right).\frac{1}{\mu_i} - \mu_i B'(\eta_i^{(2)}) - \frac{y_i}{1+\eta_i^{(2)}}$$

$$\frac{\partial^2 \ell_i}{\partial \eta_i^{(1)2}} = \left\{ -A'\left(\frac{\eta_i^{(2)}}{\mu_i}, y_i\right).\frac{\eta_i^{(2)}}{\mu_i^2} - B(\eta_i^{(2)}) + \frac{y_i}{\mu_i} \right\} \mu_i''$$
$$+ \left\{ A''\left(\frac{\eta_i^{(2)}}{\mu_i}, y_i\right).\frac{\eta_i^{(2)2}}{\mu_i^4} + A'\left(\frac{\eta_i^{(2)}}{\mu_i}, y_i\right).\frac{2\eta_i^{(2)}}{\mu_i^3} - \frac{y_i}{\mu_i^2} \right\} (\mu_i')^2$$

$$\frac{\partial^2 \ell_i}{\partial \eta_i^{(1)} \partial \eta_i^{(2)}} = \left\{ -A''\left(\frac{\eta_i^{(2)}}{\mu_i}, y_i\right).\frac{\eta_i^{(2)2}}{\mu_i^3} - A'\left(\frac{\eta_i^{(2)}}{\mu_i}, y_i\right).\frac{1}{\mu_i^2} - B'(\eta_i^{(2)}) \right\} \mu_i'$$

$$\frac{\partial^2 \ell_i}{\partial \eta_i^{(2)2}} = A''\left(\frac{\eta_i^{(2)}}{\mu_i}, y_i\right).\frac{1}{\mu_i^2} - \mu_i B''(\eta_i^{(2)}) + \frac{y_i}{(1+\eta_i^{(2)})^2}$$

where

$$A'(a,b) = \sum_{j=0}^{b-1} \frac{j}{(1+ja)}$$

$$A''(a,b) = -\sum_{j=0}^{b-1} \frac{j^2}{(1+ja)^2}$$

$$B'(a) = \begin{cases} \dfrac{a - (1+a)\log(1+a)}{a^2(1+a)} & \text{if } a \neq 0 \\ -1/2 & \text{if } a = 0 \end{cases}$$

and

$$B''(a) = \begin{cases} \dfrac{2(1+a)^2\log(1+a) - a(3a+2)}{a^3(1+a)^2} & \text{if } a \neq 0 \\ 2/3 & \text{if } a = 0 \end{cases} .$$

In practice, for numerical stability, B(a) and its derivatives should be evaluated with Taylor series round a = 0 when a is near 0,

$$B(a) = 1 - \frac{1}{2}a + \frac{1}{3}a^2 - \frac{1}{4}a^3 + \ldots.$$

$$B'(a) = -\frac{1}{2} + \frac{2}{3}a - \frac{3}{4}a^2 + \frac{4}{5}a^3 - \ldots.$$

$$B''(a) = \frac{2}{3} - \frac{2\times3}{4}a + \frac{3\times4}{5}a^2 - \frac{4\times5}{6}a^3 + \ldots$$

If b is large then A(a,b), A'(a,b) and A''(a,b) can be evaluated using the formulae

$$A(a,b) = b\log(a) + \psi(a^{-1}+b) - \psi(a^{-1})$$

$$A'(a,b) = \frac{b}{a} - \{\psi'(a^{-1}+b) - \psi'(a^{-1})\}.\frac{1}{a^2}$$

$$A''(a,b) = -\frac{b}{a^2} + \{\psi'(a^{-1}+b)-\psi'(a^{-1})\}.\frac{2}{a^3} + \{\psi''(a^{-1}+b)-\psi''(a^{-1})\}.\frac{1}{a^4}$$

where $\psi(\cdot) = \log\Gamma(\cdot)$ and $\psi'(\cdot)$ and $\psi''(\cdot)$ are the digamma and trigamma functions. Fast algorithms have been written to evaluate these functions by Bernardo(1976) and Schneider(1978).

Since there is no obvious alternative IRLS algorithm that ensures that ($\mathbf{X}'\mathbf{W}\mathbf{X}$) will be positive definite in this type of model, a definiteness modification such as that implemented by COMBINEM in

Appendix B must be available.

Good starting values cannot be easily obtained. If we use $\phi_1 = 0$ and $\phi_2 = 0$ and replace $\eta_i^{(1)}$ in (4.1.6) by its maximum likelihood estimate from the saturated model with ϕ_1 and ϕ_2 fixed at these values, $\eta_i^{(1)} = \mu^{-1}(y_i)$, then the starting values of ϕ_1 and ϕ_2 are often negative, which does not correspond to a valid distribution. The reason for this seems to be that the saturated model fits perfectly, and there is therefore no information from the fit of the saturated model to distinguish between different specifications of the $\text{var}(y_i)$. In order to get starting values such that $\eta_i^{(2)} \geq 0$ for all i, we can constrain $\phi_1 = \phi_2 = 0$ initially and get starting values for β from this Poisson model.

To illustrate the algorithm, it was applied to some artificial data. The data are assumed to be from an insecticidal experiment in which 50 plots were used, ten of which were allocated at random to each of 5 doses of the insecticide. The numbers of insects found in each plot are assumed to have means

$$E[y_i] = \exp \{ \beta_1 + \beta_2 d_i + \beta_3 d_i^2 \}$$

where d_i is the dose of insecticide for i=1,...,50; the data are shown in Table 4.5.1.

The NRL algorithm described above was applied to the data and its iterations are shown in Table 4.5.2. If the insects are randomly distributed as a Poisson process within all plots, then $\phi_1 = 0$ and $\phi_2 = 0$. In view of the two motivations for negative binomial models given by (d)(i) and (d)(ii) in Section 3.2, we would expect $\phi_1 > 0$ if the insects occur in clusters, whereas we would expect $\phi_2 > 0$ if there are systematic differences in the rate between the plots that have not been described by the model for $E[y_i]$. These constrained models were fitted using the same NRL algorithm that was used for fitting the unconstrained model, but with the constraints applied in each iteration as extra observations with infinite weights, as described in Section 3.8. For example, the iterations with the constraint $\phi_2 = 0$, which are

TABLE 4.5.1

Artificial Data for Fitting Negative Binomial Model with Variance = Mean . ($\phi_1 + \phi_2$.Mean)

Count of Insects, y_i	Dose of Insecticide, d_i	Count of Insects, y_i	Dose of Insecticide, d_i
47	0	32	0
27	0	23	0
34	0	33	0
43	0	45	0
34	0	45	0
35	1	23	1
27	1	35	1
38	1	40	1
30	1	18	1
39	1	34	1
22	2	21	2
21	2	13	2
20	2	23	2
37	2	13	2
25	2	31	2
11	3	14	3
9	3	19	3
10	3	22	3
28	3	12	3
13	3	12	3
1	4	1	4
10	4	7	4
3	4	3	4
4	4	6	4
7	4	7	4

also NR iterations, are also shown in Table 4.5.2.

The deviances which are defined as minus twice the kernel of the log-likelihoods, $-2 \sum \ell_i$, for the four models are

unconstrained	:	4989.57
$\phi_1 = 0$	:	4986.81
$\phi_2 = 0$	:	4989.35
$\phi_1 = \phi_2 = 0$	:	4975.80

TABLE 4.5.2

Iterations of the NRL Algorithm Applied to Data in Table 4.5.1. Starting Values (Iteration 0) were Found as Decribed in the Text.

Iteration	β_1	β_2	β_3	ϕ_1	ϕ_2	Log-Likelihood	Convergence Rate, C
Unconstrained							
0	3.60288	0.01566	-0.10940	0.00000	0.00000	2486.00219	
1	3.58283	0.02681	-0.11904	0.34118	-0.00095	2492.52224	0.73156
2	3.57836	0.02892	-0.12177	0.75450	-0.00531	2494.36883	0.59596
3	3.57909	0.02677	-0.12117	1.12382	-0.01212	2494.74917	0.38196
4	3.57975	0.02481	-0.12026	1.31548	-0.01703	2494.78415	0.14708
5	3.57985	0.02443	-0.12008	1.34767	-0.01797	2494.78474	0.02024
6	3.57986	0.02442	-0.12008	1.34832	-0.01799	2494.78474	0.00116
7	3.57986	0.02442	-0.12008	1.34832	-0.01799	2494.78474	0.00000
Constrained ϕ_2=0							
0	3.60288	0.01566	-0.10940	0.00000	0.00000	2486.00219	
1	3.58275	0.02782	-0.11941	0.32035	0.00000	2492.50640	0.63704
2	3.57762	0.03274	-0.12305	0.64104	0.00000	2494.31240	0.47792
3	3.57795	0.03365	-0.12354	0.86148	0.00000	2494.65351	0.27108
4	3.57858	0.03359	-0.12342	0.93637	0.00000	2494.67565	0.08203
5	3.57865	0.03357	-0.12340	0.94298	0.00000	2494.67579	0.00698
6	3.57865	0.03357	-0.12340	0.94303	0.00000	2494.67579	0.00005

Since the difference in deviance caused by constraining one parameter should have approximately a χ^2 distribution with one degree of freedom, the conclusion with this artificial example would be that there was strong evidence that the Poisson distribution did not fit the data, but that there was insufficient data to distinguish between whether that has been caused by clustering alone ($\phi_2 = 0$) or by unexplained systematic differences between the means in the plots, or a mixture of these effects. The deviance for the model with $\beta_3 = 0$ but ϕ_1 and ϕ_2 unconstrained is 4973.34, so that a log-linear model for $E[y_i]$ without a quadratic term is clearly inadequate in this example.

4.6 EXAMPLES : ROBUST ESTIMATION IN LINEAR MODELS

In this section, we consider maximum likelihood estimation in models with log probability density functions

$$\ell_i(\eta_i, \sigma^2) = -\phi\left(\frac{y_i - \eta_i}{\sigma}\right) - \log \sigma - \log\left\{ \int_{-\infty}^{\infty} \exp(-\phi(z))dz \right\} \qquad (4.6.1)$$

where $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$, $\phi(\cdot)$ is a symmetric function and the distribution is longer-tailed than the normal distribution. Several such models were described in Section 3.2(f). Most models have an extra shape parameter ϕ. In practice, this is usually fixed by the user to give the desired degree of robustness and we shall initially treat any such ϕ as a known constant. Later in the section, we shall also consider estimation of ϕ.

The two systematic parts $\mathbf{x}_i'\boldsymbol{\beta}$ and $\log \sigma$ (or $\mathbf{x}_i'\boldsymbol{\beta}$ and any other function of σ^2) are detached provided $\phi(\cdot)$ is symmetric. Alternation between FS for $\boldsymbol{\beta}$ with σ^2 fixed and FS for $\log \sigma$ with $\boldsymbol{\beta}$ held fixed, is therefore a minor improvement over joint FS. NRL is equivalent to NR and NRL-alternation would also be expected to be reasonably efficient.

The various methods are illustrated using the log-tailed distribution with

$$\phi(u) = \begin{cases} u^2/2 & u < \phi \\ \phi^2 \log(|u|/\phi) + \phi^2/2 & u \geq \phi \end{cases} \qquad (4.6.2)$$

For the NRL-alternation algorithm, the IRLS formulae for $\boldsymbol{\beta}$ with fixed σ^2 have "explanatory" variables $\mathbf{x}_i$, weights

$$w_i = \begin{cases} \sigma^{-2} & |y_i - \mathbf{x}_i'\boldsymbol{\beta}|/\sigma < \phi \\ -\dfrac{\phi^2}{(y_i - \mathbf{x}_i'\boldsymbol{\beta})^2} & |y_i - \mathbf{x}_i'\boldsymbol{\beta}|/\sigma \geq \phi \end{cases}$$

and responses

$$z_i = \begin{cases} (y_i - \mathbf{x}_i'\boldsymbol{\beta}) & |y_i - \mathbf{x}_i'\boldsymbol{\beta}|/\sigma < \phi \\ -(y_i - \mathbf{x}_i'\boldsymbol{\beta}) & |y_i - \mathbf{x}_i'\boldsymbol{\beta}|/\sigma \geq \phi \end{cases} .$$

The systematic part $\mathbf{x}_i'\boldsymbol{\beta}$ is therefore not convex and a definiteness modification may be needed. The NR step for $(\log \sigma)$ with $\boldsymbol{\beta}$ assumed fixed can be written as a least squares calculation, but it is more conveniently expressed as

$$\log \sigma^+ = \log \sigma + \frac{\Sigma\, a_i - n}{\Sigma\, b_i} \qquad (4.6.3)$$

where
$$a_i = \begin{cases} \dfrac{(y_i-\mathbf{x}_i'\boldsymbol{\beta})^2}{\sigma^2} & |y_i-\mathbf{x}_i'\boldsymbol{\beta}|/\sigma < \phi \\ \phi^2 & |y_i-\mathbf{x}_i'\boldsymbol{\beta}|/\sigma \geq \phi \end{cases}$$

$$b_i = \begin{cases} \dfrac{2(y_i-\mathbf{x}_i'\boldsymbol{\beta})^2}{\sigma^2} & |y_i-\mathbf{x}_i'\boldsymbol{\beta}|/\sigma < \phi \\ 0 & |y_i-\mathbf{x}_i'\boldsymbol{\beta}|/\sigma \geq \phi \end{cases}$$

If FS-alternation is used rather than NRL-alternation, the IRLS weights are

$$w_i^* = \sigma^{-2} \frac{\int_{-\infty}^{\infty} \Phi''(u) \exp\{-\Phi(u)\}\, du}{\int_{-\infty}^{\infty} \exp\{-\Phi(u)\}\, du}$$

$$= \sigma^{-2} \frac{\int_0^{\phi} e^{-u^2/2}\, du - \int_{\phi}^{\infty} \phi^2 u^{-2}\, \phi^{\phi^2}\, u^{-\phi^2}\, e^{-\phi^2/2}\, du}{\int_0^{\phi} e^{-u^2/2}\, du + \int_{\phi}^{\infty} \phi^{\phi^2}\, u^{-\phi_2}\, e^{-\phi^2/2}\, du}$$

$$= \sigma^{-2} \frac{\sqrt{(2\pi)}\ (\Phi(\phi)-0.5) - \phi^{(\phi^2+2)}\, e^{-\phi^2/2} \int_{\phi}^{\infty} u^{-(\phi^2+2)}\, du}{\sqrt{(2\pi)}\ (\Phi(\phi)-0.5) + \phi^{\phi^2}\, e^{-\phi^2/2} \int_{\phi}^{\infty} u^{-\phi^2}\, du}$$

$$= \sigma^{-2} \frac{\sqrt{(2\pi)}\ (\Phi(\phi)-0.5) - \phi^{(\phi^2+2)}\, e^{-\phi^2/2}\, \phi^{-(\phi^2+1)} / (\phi^2+1)}{\sqrt{(2\pi)}\ (\Phi(\phi)-0.5) + \phi^{\phi^2}\, e^{-\phi^2/2}\, \phi^{-\phi^2+1} / (\phi^2-1)}$$

$$= \sigma^{-2} \frac{\sqrt{(2\pi)}\ (\Phi(\phi)-0.5) - \phi\, e^{-\phi^2/2} / (\phi^2+1)}{\sqrt{(2\pi)}\ (\Phi(\phi)-0.5) + \phi\, e^{-\phi^2/2} / (\phi^2-1)}$$

and the "responses" are

$$z_i = \begin{cases} \dfrac{(y_i - \mathbf{x}_i'\boldsymbol{\beta})}{w_i^* \sigma^2} & |y_i - \mathbf{x}_i'\boldsymbol{\beta}|/\sigma < \phi \\ \dfrac{\phi^2}{w_i^*(y_i - \mathbf{x}_i'\boldsymbol{\beta})} & |y_i - \mathbf{x}_i'\boldsymbol{\beta}|/\sigma \geq \phi \end{cases} .$$

To obtain the FS step for $\log \sigma$ with $\boldsymbol{\beta}$ fixed, all b_i in (4.6.3) must be replaced by their expected value,

$$b = E[b_i] = \frac{\int_0^\phi 2\, u^2\, e^{-u^2/2}\, du}{\int_0^\phi e^{-u^2/2}\, du + \int_\phi^\infty \phi^2\, u^{-\phi^2}\, e^{-\phi^2/2}\, du}$$

$$= \frac{2\sqrt{2} \int_0^{\phi^2/2} v^{1/2}\, e^{-v}\, dv}{\sqrt{(2\pi)}\,(\Phi(\phi) - 0.5) + \phi\, e^{-\phi^2/2} / (\phi^2 - 1)}$$

where the numerator is an incomplete gamma function and can be evaluated by algorithms such as Moore (1982). When $\phi = 1.3$, $w_i^* = 0.44125624\sigma^{-2}$ and $b = 0.49697928$.

The full NRL algorithm based on systematic parts $\mathbf{x}_i'\boldsymbol{\beta}$ and $\log \sigma$, which is also NR for $\boldsymbol{\beta}$ and $\log \sigma$, can be expressed as a weighted least squares calculation using (4.1.1). Since some weights for that calculation are zero if any $|y_i - \mathbf{x}_i'\boldsymbol{\beta}|/\sigma > \phi$, the corresponding rows of data must be replaced by two pseudo-observations, as described by (3.4.3). It is easier to implement NRL by evaluating the derivatives

$$\frac{\partial \ell}{\partial \log \sigma} = \sum a_i - n$$

$$\frac{\partial^2 \ell}{\partial (\log \sigma)^2} = -\sum b_i$$

and

$$\frac{\partial^2 \ell}{\partial \boldsymbol{\beta}\, \partial (\log \sigma)} = -\sum c_i \mathbf{x}_i$$

where

$$c_i = \begin{cases} \dfrac{2(y_i - \mathbf{x}_i'\boldsymbol{\beta})}{\sigma^2} & |y_i - \mathbf{x}_i'\boldsymbol{\beta}|/\sigma < \phi \\ 0 & |y_i - \mathbf{x}_i'\boldsymbol{\beta}|/\sigma \geq \phi \end{cases}$$

and to use subroutine NUISNC to incorporate parameter ($\log \sigma$) into the NR iteration for $\boldsymbol{\beta}$ with σ^2 fixed.

TABLE 4.6.1

Data From Daniel and Wood (1971, Chapter 5)

y_i	x_{i2}	x_{i3}	x_{i4}
42	80	27	89
37	80	27	88
37	75	25	90
28	62	24	87
18	62	22	87
18	62	23	87
19	62	24	93
20	62	24	93
15	58	23	87
14	58	18	80
14	58	18	89
13	58	17	88
11	58	18	82
12	58	19	93
8	50	18	89
7	50	18	86
8	50	19	72
8	50	19	79
9	50	20	80
15	56	20	82
15	70	20	91

Andrews (1974) showed that robust estimation of the parameters in an example with

$$E[y_i] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} \qquad i=1,\ldots,21$$

showed up four outliers that had only been found after detailed analysis by Daniel and Wood(1971, chapter 5). The data are given in Table 4.6.1. Andrews estimated the parameters to maximize a function that does not correspond to any log-likelihood. Stirling(1984) showed

that maximum likelihood estimation of β and σ^2 in the log-tail distribution (4.6.2) with $\phi=1.3$ gives $\hat{\beta}_0=-37.73$, $\hat{\beta}_1=0.8529$, $\hat{\beta}_2=0.4610$, $\hat{\beta}_3=-0.07372$ and $\hat{\sigma}=0.7796$. This results in residuals which show up the four outliers in essentially the same way as Andrews' method. Tables 4.6.2 and 4.6.3 show the successive iterations of the NRL-alternation, FS-alternation and NR algorithms described above, each starting from the least squares estimate $\hat{\beta}_0$ of β and $\hat{\sigma}_0^2 = \Sigma(y_i-\mathbf{x}_i'\hat{\beta}_0)^2/n$. (Alternative starting values would have been the $\phi(u)=|u|$ estimates for which there is a linear programming solution). In iterations 2 to 5 of the NR algorithm ($\mathbf{X}'\mathbf{W}\mathbf{X}$) was found to be not positive definite and a Gill and Murray (1974) definiteness modification was applied with the diagonal of the second derivative matrix subsequently increased by ten percent; in iterations 1 and 6, a stepsize modification was needed to increase the likelihood. Stepsizes were not otherwise improved by interpolation or extrapolation in any iterations. In the NRL-alternation algorithm, definiteness modifications were needed in iterations 2 and 3. The FS-alternation algorithm had extremely poor convergence, but needed no stepsize or definiteness modifications.

If the NRL-alternation or FS-alternation algorithms are used, the variance-covariance matrix of $\hat{\beta}$ can be estimated by $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ from the last IRLS iteration for β. Log $\hat{\sigma}$ is asymptotically independent and its variance can be estimated as $(\sum b_i)^{-1}$ or $(nb)^{-1}$ for the two algorithms. The variances can also be consistently estimated with $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ from the last iteration of the NR algorithm.

We next extend the methods to permit maximum likelihood estimation of an additional parameter ϕ which affects the amount of robustness in the estimator. Such estimators are called <u>adaptive</u>. We therefore now consider models of the form

$$\ell_i(\eta_i,\sigma^2,\phi) = -\phi\left(\frac{y_i-\eta_i}{\sigma},\phi\right) - \log\sigma - \log\left(\int_{-\infty}^{\infty}\exp\{-\phi(z,\phi)\}dz\right) \tag{4.6.4}$$

where $\phi(\cdot,\cdot)$ is symmetric round zero in its first parameter. The

TABLE 4.6.2

Iterations of NR and NRL-alternation Applied to Fit Log-tail Model with $\phi = 1.3$ to Data in Table 4.6.1. Starting Values (Iteration 0) were Found from Ordinary Least Squares.

Iteration	β_0	β_1	β_2	β_3	log σ	Log-Likelihood	Convergence Rate, C
NR algorithm							
0	-39.91967	0.71564	1.29529	-0.15212	1.07096	-31.41310	
1	(-58.12608	5.11831	-14.55171	0.83600	-8.36399	-224.42999)	
	(-44.47128	1.81631	-2.66646	0.09491	-1.28778	-76.68553)	
	-41.05757	0.99081	0.30485	-0.09036	0.48127	-26.80134	0.95809
2	-40.73566	0.98243	0.31085	-0.08928	0.34271	-26.23380	0.89099
3	-40.39209	0.97290	0.31506	-0.08846	0.25016	-25.83965	0.88414
4	-40.04462	0.96303	0.31891	-0.08744	0.15481	-25.44973	0.86696
5	-39.73446	0.95466	0.32224	-0.08628	0.06338	-25.12434	0.86100
6	(-24.32807	-0.36184	1.24220	0.39313	-6.31030	-146.65804)	
	(-35.88287	0.62553	0.55223	0.03358	-1.53004	-34.37459)	
	-38.73468	0.86923	0.38194	-0.05517	-0.35023	-23.46064	0.47409
7	-37.29614	0.84300	0.46858	-0.07348	-0.23512	-23.05211	0.38475
8	-37.75755	0.85384	0.46015	-0.07391	-0.26033	-23.00667	0.08125
9	-37.73323	0.85291	0.46099	-0.07373	-0.24922	-23.00570	0.01764
10	-37.73281	0.85288	0.46101	-0.07372	-0.24902	-23.00570	0.00031
11	-37.73281	0.85288	0.46101	-0.07372	-0.24902	-23.00570	0.00000
NRL-alternation							
0	-39.91967	0.71564	1.29529	-0.15212	1.07096	-31.41310	
1	-42.26718	0.90546	0.72347	-0.11581		-30.43166	
					0.35054	-26.93374	1.20504
2	-41.80167	0.89806	0.69554	-0.11300		-26.39550	
					0.32123	-26.39089	0.90294
3	-41.07095	0.88682	0.66209	-0.10771		-25.80102	
					0.21419	-25.70516	0.82095
4	-36.80414	0.83546	0.51096	-0.08528		-23.97102	
					-0.13755	-23.14717	0.27494
5	-37.59693	0.84997	0.46890	-0.07527		-23.06943	
					-0.25533	-23.00984	0.13809
6	-37.74077	0.85303	0.46058	-0.07362		-23.00593	
					-0.24934	-23.00571	0.05788
7	-37.73319	0.85289	0.46099	-0.07372		-23.00570	
					-0.24904	-23.00570	0.04770
8	-37.73283	0.85288	0.46101	-0.07372		-23.00570	
					-0.24902	-23.00570	0.04349

systematic part $(\mathbf{x}_i'\boldsymbol{\beta})$ is detached from the pair of systematic parts (σ, ϕ) in this type of model since

$$E\left[\frac{\partial^2 \ell_i(\eta_i, \sigma, \phi)}{\partial \eta_i \, \partial \sigma}\right] = E\left[\frac{\partial^2 \ell_i(\eta_i, \sigma, \phi)}{\partial \eta_i \, \partial \phi}\right] = 0 \quad ,$$

so that alternation between iterations for $\boldsymbol{\beta}$ with (σ, ϕ) fixed and iterations for (σ, ϕ) with $\boldsymbol{\beta}$ fixed could be used. For example, in the t-distribution, which has $\Phi(\cdot, \cdot)$ defined by

$$\Phi(u, \phi) = (\phi + 1)/2 \cdot \log(1 + u^2/\phi) \quad ,$$

the weights and "responses" for the NRL IRLS iterations for $\boldsymbol{\beta}$ with fixed (σ, ϕ) are

$$w_i = \frac{\phi+1}{\sigma^2} \cdot \frac{(\phi - (y_i - \mathbf{x}_i'\boldsymbol{\beta})^2/\sigma^2)}{(\phi + (y_i - \mathbf{x}_i'\boldsymbol{\beta})^2/\sigma^2)^2}$$

and

$$z_i = \frac{(y_i - \mathbf{x}_i'\boldsymbol{\beta}) \cdot (\phi + (y_i - \mathbf{x}_i'\boldsymbol{\beta})^2/\sigma^2)}{(\phi - (y_i - \mathbf{x}_i'\boldsymbol{\beta})^2/\sigma^2)}$$

respectively. The NR iterations for the two systematic parts, $\log \sigma$ and ϕ, with fixed $\boldsymbol{\beta}$, can be expressed as least squares calculations using (4.1.1) and the derivatives,

$$\frac{\partial \ell_i}{\partial \log \sigma} = \frac{\phi\,(e_i^2 - 1)}{(e_i^2 + \phi)}$$

$$\frac{\partial^2 \ell_i}{\partial (\log \sigma)^2} = -2\,\phi(\phi+1)\frac{e_i^2}{(e_i^2+\phi)^2}$$

$$\frac{\partial \ell_i}{\partial \phi} = -\frac{1}{2}\log\left(1+\frac{e_i^2}{\phi}\right) + \frac{1}{2}\frac{(e_i^2-1)}{(e_i^2+\phi)} + \frac{1}{2}\left(\psi'\left(\frac{\phi+1}{2}\right) - \psi'\left(\frac{\phi}{2}\right)\right)$$

$$\frac{\partial^2 \ell_i}{\partial \phi^2} = \frac{1}{2}\frac{(e_i^4+\phi)}{\phi\,(e_i^2+\phi)^2} + \frac{1}{4}\left(\psi''\left(\frac{\phi+1}{2}\right) - \psi''\left(\frac{\phi}{2}\right)\right)$$

and

$$\frac{\partial^2 \ell_i}{\partial \phi \, \partial \log \sigma} = \frac{e_i^2\,(e_i^2 - 1)}{(e_i^2 + \phi)^2}$$

where $e_i = (y_i - \mathbf{x}_i'\boldsymbol{\beta})/\sigma$, $\psi(\cdot) = \log \Gamma(\cdot)$ and $\psi'(\cdot)$ and $\psi''(\cdot)$ are its

TABLE 4.6.3

Iterations of FS-alternation Applied to Fit Log-tail Model with $\phi = 1.3$ to Data in Table 4.6.1. Starting Values (Iteration 0) were Found from Ordinary Least Squares.

Iteration	β_0	β_1	β_2	β_3	log σ	Log-Likelihood	Convergence Rate, C
FS-alternation							
0	-39.91967	0.71564	1.29529	-0.15212	1.07096	-31.41310	
1	-41.68171	0.94721	0.50367	-0.09949		-30.64816	
					0.04896	-26.36211	0.96772
2	-34.84187	0.91549	0.47756	-0.15160		-24.64215	
					0.06620	-24.64130	0.76269
3	-39.18098	0.86384	0.49573	-0.07282		-23.65608	
					-0.16544	-23.30853	0.46929
4	-36.31665	0.85928	0.45571	-0.09294		-23.12436	
					-0.22450	-23.10673	0.93239
5	-38.70211	0.85191	0.47044	-0.06438		-23.06377	
					-0.24680	-23.06138	0.67359
6	-36.92860	0.85417	0.45639	-0.08260		-23.04034	
					-0.23614	-23.03982	0.83921
7	-38.33739	0.85186	0.46570	-0.06732		-23.02735	
					-0.25034	-23.02639	0.74289
8	-37.24715	0.85352	0.45807	-0.07895		-23.01875	
					-0.24251	-23.01847	0.81067
9	-38.10493	0.85227	0.46370	-0.06975		-23.01380	
					-0.25079	-23.01348	0.76079
10	-37.43697	0.85328	0.45916	-0.07690		-23.01061	
					-0.24558	-23.01049	0.79944
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
27	-37.73738	0.85287	0.46104	-0.07367		-23.00571	
					-0.24906	-23.00571	0.78253
28	-37.72924	0.85289	0.46099	-0.07376		-23.00571	
					-0.24899	-23.00571	0.78299
29	-37.73561	0.85288	0.46103	-0.07369		-23.00570	
					-0.24905	-23.00570	0.78263
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:

first and second derivatives. In this example however, there is little advantage in implementing the NR iteration for log σ and ϕ using least squares with (4.1.1) over direct evaluation of the NR step from the derivatives by directly inverting the second derivative matrix.

In this problem, the expected second derivatives only depend on ϕ. They cannot however be explicitly found, so that the three expected second derivatives must be reevaluated by a numerical integration in each iteration. FS does not therefore seem a practical algorithm here.

Starting values for β and σ^2 can be found as described earlier for models with fixed ϕ. A starting value for ϕ is harder to obtain. A moderate value such as $\phi = 5.0$ is often suitable, though an initial estimate from the kurtosis of the least squares residuals would also be possible.

TABLE 4.6.4

Artificial Data Generated from a T-distribution
With 3 Degrees of Freedom (Dempster et al., 1980)

-0.141	0.678
-0.036	-0.350
-5.005	0.886
0.485	-4.154
1.415	1.546

To illustrate the NRL-alternation algorithm, it was applied to an artificial data set that was generated from the t-distribution with 3 degrees of freedom and mean zero. The data are shown in Table 4.6.4 and were used by Dempster et al.(1980) to illustrate another type of maximum likelihood algorithm called the EM algorithm; this latter algorithm is discussed in detail in Chapter 6. The iterations of the NRL-alternation algorithm are given in Table 4.6.5. It converges well

apart from a stepsize modification that in needed in iteration 2.

TABLE 4.6.5

Iterations of NRL-alternation Applied to Fit a T-distribution to Data in Table 4.6.1. Starting Values (Iteration 0) were $\phi = 5.0$ and the Ordinary Least Squares Estimates of μ and σ^2

Iteration	Mean, μ	log σ	ϕ	Log-Likelihood	Convergence Rate, C
0	-0.46760	0.76450	5.00000	-16.37611	
1	-0.06923	0.36187	2.49408	-16.23665 -15.57609	0.41504
2	0.29996	(-2.23081	-9.54926	-15.33955 $-\infty$)	
		(-0.28630	-0.51675	$-\infty$)	
3	0.36714	0.19983	1.74138	-14.98577 -14.97385	0.45279
4	0.44956	-0.35171	0.96687	-14.62523 -14.59898	0.43453
5	0.43705	-0.29287	1.16734	-14.49432 -14.49366	0.33345
6	0.43317	-0.26778	1.25944	-14.48237 -14.48231	0.15753
7	0.43257	-0.26371	1.27450	-14.48209 -14.48209	0.05126
8	0.43254	-0.26355	1.27494	-14.48209 -14.48209	0.02239
		-0.26355	1.27495	-14.48209	0.01298

5. ELIMINATION AND ADJUSTMENT OF PARAMETERS

5.1 GENERAL DESCRIPTION OF ELIMINATION AND ADJUSTMENT

If there are many unknown parameters β in a model, optimization algorithms such as NR, NRL and FS can be very slow and may have difficulty converging. If the parameters are partitioned $\beta' = [\beta_1' \ \beta_2']$ and if $\hat{\beta}_1(\beta_2)$ can easily be found to maximize the log-likelihood $\ell(\beta_1, \beta_2)$ with respect to β_1 for fixed β_2, then β_1 can be eliminated from the log-likelihood function, after which the optimization can be replaced by the lower-dimensioned problem of maximizing $\ell(\hat{\beta}_1(\beta_2), \beta_2)$. This reduction in dimension usually means that standard iterative optimization algorithms are less likely to diverge, modifications to ensure convergence are easier to apply, and considerably fewer iterations are often required. Golub and Pereyra (1973), Kaufman (1975) and Spitzer (1982) give numerical illustrations of the potential improvements. Elimination was first proposed in this general form by Richards (1961), but was also suggested for the special problem of nonlinear least squares estimation by Koopmans and Hood (1953), Lawton and Sylvestre (1971) and Guttman et al (1973), all apparently independently. In this chapter we investigate the technique of elimination and a related technique called adjustment.

If there are only one or two parameters in β_2 then the easiest way to make use of the eliminated likelihood is to evaluate $\ell(\hat{\beta}_1(\beta_2), \beta_2)$ for various β_2 and plot these values against β_2. For example, Box and Cox (1964) proposed a model where transformations of the measured responses, $y_i^{(\lambda)}$ for $i=1,\ldots,n$, satisfy ordinary linear model assumptions. They eliminated the linear parameters and the error variance and used a plot of the eliminated log-likelihood against λ to obtain a confidence interval for λ. Clayton (1983) suggested a similar

graphical approach for a class of two sample survival data models. In the rest of this chapter, we consider numerical optimization methods.

There are two ways in which general optimization algorithms can be modified to make use of the relationship, $\hat{\beta}_1 = \hat{\beta}_1(\beta_2)$. The algorithms can either be applied directly to the eliminated log-likelihood $\ell(\hat{\beta}_1(\beta_2), \beta_2)$ or they can be applied to $\ell(\beta_1, \beta_2)$ with an adjustment of β_1 to $\hat{\beta}_1(\beta_2)$ between iterations; we call these elimination and adjustment algorithms, respectively. Clearly adjustment always improves an iteration since it increases the likelihood.

In Section 5.2 we investigate the relationship between elimination and adjustment when these are applied to the NR, NRL and FS algorithms. Elimination and adjustment are particularly helpful for nonlinear least squares and Section 5.3 discusses this application. A nonlinear least squares example is used to give a numerical comparison of the methods in Section 5.4. Other applications where elimination and adjustment can be used are discussed in Section 5.5. Finally, related methods of improving the NRL and FS algorithms for models whose systematic parts contain linear and nonlinear parameters are investigated in Section 5.6.

5.2 ELIMINATION AND ADJUSTMENT APPLIED TO THE NR, NRL AND FS ALGORITHMS

In a full Newton-Raphson (NR) iteration the value of $\boldsymbol{\beta}$ is updated to

$$\boldsymbol{\beta}^{+} = \boldsymbol{\beta} - \left(\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2} \right)^{-1} \left(\frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \right) . \qquad (5.2.1)$$

Similarly a NR-elimination iteration for $\boldsymbol{\beta}_2$ is defined to be

$$\boldsymbol{\beta}_2^{+} = \boldsymbol{\beta}_2 - \left(\frac{\partial^2 \ell(\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2), \boldsymbol{\beta}_2)}{\partial \boldsymbol{\beta}_2^2} \right)^{-1} \left(\frac{\partial \ell(\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2), \boldsymbol{\beta}_2)}{\partial \boldsymbol{\beta}_2} \right) . \qquad (5.2.2)$$

The eliminated log-likelihood $\ell(\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2), \boldsymbol{\beta}_2)$ is usually a complex function of $\boldsymbol{\beta}_2$. However its derivatives can be easily found using the following results from Richards (1961).

Theorem 5.2.1

$$\frac{\partial \ell(\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2), \boldsymbol{\beta}_2)}{\partial \boldsymbol{\beta}_2} = \left[\frac{\partial \ell(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)}{\partial \boldsymbol{\beta}_2} \right]_{\boldsymbol{\beta}_1 = \hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)} , \text{ and}$$

$$\frac{\partial^2 \ell(\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2), \boldsymbol{\beta}_2)}{\partial \boldsymbol{\beta}_2^2} = \left[\frac{\partial^2 \ell(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)}{\partial \boldsymbol{\beta}_2^2} - \left(\frac{\partial^2 \ell(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)}{\partial \boldsymbol{\beta}_1 \partial \boldsymbol{\beta}_2} \right)' \left(\frac{\partial^2 \ell(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)}{\partial \boldsymbol{\beta}_1^2} \right)^{-1} \left(\frac{\partial^2 \ell(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)}{\partial \boldsymbol{\beta}_1 \partial \boldsymbol{\beta}_2} \right) \right]_{\boldsymbol{\beta}_1 = \hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)}$$

Once $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ has been evaluated, the calculations required to evaluate $\boldsymbol{\beta}_2^{+}$ using (5.2.2) and the Theorem involve the same derivatives as those needed for a joint NR iteration on $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ and smaller matrices need to be inverted. A NR elimination step will be as fast as a joint NR step provided the additional time required to evaluate $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ is small.

If β_1 is adjusted to $\hat{\beta}_1(\beta_2)$ before a joint NR iteration, so that $\partial \ell(\beta)/\partial \beta_1 = 0$, and the second derivative matrix in (5.2.1) is partitioned, it can be shown using Theorem 5.2.1 that the resulting step for β_2 is identical to that given by (5.2.2). Therefore the NR elimination and NR adjustment methods are identical.

The matrix $[\partial^2 \ell(\beta_1, \beta_2)/\partial \beta_1^2]^{-1}$ is usually not evaluated explicitly. If the Choleski factorization

$$\frac{\partial^2 \ell(\beta_1, \beta_2)}{\partial \beta_1^2} = R_{11}' R_{11} \tag{5.2.3}$$

is found, where R_{11} is upper triangular, then

$$R_{11}' U = [\partial^2 \ell(\beta_1, \beta_2)/\partial \beta_1 \partial \beta_2]$$

can be solved for U and

$$\frac{\partial^2 \ell(\hat{\beta}_1(\beta_2), \beta_2)}{\partial \beta_2^2} = \left[\frac{\partial^2 \ell(\beta_1, \beta_2)}{\partial \beta_2^2} \right]_{\beta_1 = \hat{\beta}_1(\beta_2)} - U'U \quad .$$

This method is similar to one described by Spitzer (1982).

Elimination and adjustment can also be applied to the NRL algorithm. We first examine models with a single systematic part and log-likelihood components $\ell_i(\eta_i)$ where $\eta_i = \eta_i(\beta_1, \beta_2)$. Then, writing

$$X_1 = \partial \eta / \partial \beta_1 \text{ and } X_2 = \partial \eta / \partial \beta_2 \tag{5.2.4}$$

and denoting the diagonal matrix with i'th diagonal element $(-\partial^2 \ell_i / \partial \eta_i^2)$ as W, the NRL-alternation algorithm has the same form as NR-alternation, but with $\partial^2 \ell(\beta_1, \beta_2)/\partial \beta_i \partial \beta_j$ replaced by $(X_i' W X_j)$ for $i=1,2$ and $j=1,2$. The NRL-adjustment iteration can be evaluated accurately using similar calculations to those for joint NRL. The joint NRL iteration is usually found by applying a QR algorithm to $[X_1 : X_2 : z]$ with weights W where z is defined in (3.4.1). If the associated Choleski factor R is partitioned,

$$\mathbf{R} = \begin{Bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{r}_{21} \\ \mathbf{0} & \mathbf{R}_{22} & \mathbf{r}_{22} \\ \mathbf{0} & \mathbf{0} & r_3 \end{Bmatrix} \tag{5.2.5}$$

then $\boldsymbol{\beta}_2^+$ is found by solving

$$\mathbf{R}_{22}(\boldsymbol{\beta}_2^+ - \boldsymbol{\beta}_2) = \mathbf{r}_{22}$$

by back-substitution and then solving

$$\mathbf{R}_{11}(\boldsymbol{\beta}_1^+ - \boldsymbol{\beta}_1) = \mathbf{r}_{21} - \mathbf{R}_{12}(\boldsymbol{\beta}_2^+ - \boldsymbol{\beta}_2) \tag{5.2.6}$$

by back-substitution. The NRL-adjustment algorithm finds $\boldsymbol{\beta}_2^+$ in exactly the same way, but evaluates $\boldsymbol{\beta}_1^+$ from $\boldsymbol{\beta}_1^+ = \hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2^+)$. NRL-adjustment will be as fast as joint NRL if evaluating $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ is as fast as solving (5.2.6). In some types of model such as nonlinear least squares models, evaluation of $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ can be incorporated into the evaluation of $\mathbf{R}$ in the next iteration and NRL-adjustment is faster than joint NRL. In models with two (or more) systematic parts, NRL-adjustment can be implemented in a similar way, but with $\mathbf{X}_1$ and $\mathbf{X}_2$ defined by partitioning the "explanatory" variables in (4.1.1).

The NRL-elimination iterations are based on rewriting the systematic part as $\eta_i = \eta_i(\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2), \boldsymbol{\beta}_2)$ and can be expressed as IRLS calculations with the same weights and responses as for joint NRL, but explanatory variables

$$\mathbf{x}_i = \partial \eta_i(\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2), \boldsymbol{\beta}_2) \,/\, \partial \boldsymbol{\beta}_2 \tag{5.2.7}$$

In many problems, this $\mathbf{x}_i$ cannot be easily evaluated and NRL-elimination iterations are therefore usually slower than NRL-adjustment iterations. When the systematic part η_i is nonlinear, NRL-elimination and NRL-adjustment are usually not equivalent. In models with two or more systematic parts, NRL-elimination can be defined in a similar way.

Another commonly used alternative to NR is Fisher's scoring technique (FS) in which the second derivative matrix in (5.2.1) is replaced by its expected value. The FS-adjustment iterations have the

same form as NR-elimination/adjustment iterations, but with all second derivatives replaced by their expected values. Like joint FS iterations, they are always in an ascent direction. FS-elimination is usually distinct from FS-adjustment and is rarely practical since there are no general formulae corresponding to those in Theorem 5.2.1 for evaluating $E[\partial^2 \ell(\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2), \boldsymbol{\beta}_2)/\partial \boldsymbol{\beta}_2^2]$, and also the expectation usually involves $\boldsymbol{\beta}_1$ as well as $\boldsymbol{\beta}_2$.

Since adjustment increases the likelihood in an iteration, NR-elimination/adjustment is less likely to diverge than joint NR and, when both algorithms converge, considerably fewer iterations are often required. NRL-adjustment and FS-adjustment have similar advantages over joint NRL and joint FS. However convergence problems are often encountered with all algorithms, especially if poor starting values are used. Firstly, in NR algorithms the matrix of second derivatives may not be negative definite and the iterations may then approach a minimum or saddle point of the likelihood. Since $\partial^2 \ell(\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2), \boldsymbol{\beta}_2)/\partial \boldsymbol{\beta}_2^2$ is negative definite whenever $\partial^2 \ell(\boldsymbol{\beta})/\partial \boldsymbol{\beta}^2$ is negative definite, this problem is more common with joint NR than with NR-elimination/adjustment. A similar result holds for the NRL algorithms. If the problem arises, a definiteness modification must be used. This type of modification is easier and faster for NR elimination/adjustment than for joint NR since the second derivative matrix is smaller in the former iterations. For example, if $\boldsymbol{\beta}_2$ is a single parameter, the second derivative of the eliminated log-likelihood can be replaced by minus its absolute value to get an ascent direction. Similarly, for NRL-adjustment, a definiteness modification need only be applied to $\mathbf{R}_{22}$ in (5.2.5) rather then to

$$\mathbf{R}_1 = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix}$$

and is therefore easier to apply. This type of problem does not arise with joint FS or FS-adjustment.

A second problem that may be encountered with all algorithms is that an iteration may take too big a step which reduces the likelihood.

If this happens, a stepsize modification must be used. The problem is less likely for adjustment algorithms than for the corresponding joint algorithms since adjustment increases the likelihood. However adjustment and elimination algorithms must reevaluate $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ each time a new step for $\boldsymbol{\beta}_2$ is tried so that step reduction is usually slower than in the corresponding joint algorithms.

The choice of whether or not to use elimination or adjustment would be based on whether the reduction in the number of iterations is outweighed by the extra time required to evaluate $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ in each iteration. The algorithms are compared in more detail for the problem of nonlinear least squares in Section 5.3.

To conclude this section, we note that the inverse of $-\partial^2\ell(\boldsymbol{\beta})/\partial\boldsymbol{\beta}^2$ can be used to consistently estimate the variance of the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ (Sections 3.8 and 4.3). If NR-elimination/adjustment is used, this corresponds to estimating $\text{var}(\hat{\boldsymbol{\beta}}_2)$ with $\mathbf{V}_{22} = -[\partial^2\ell(\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2),\boldsymbol{\beta}_2)/\partial\boldsymbol{\beta}_2^2]^{-1}$ from the last iteration of the algorithm. The covariance between $\hat{\boldsymbol{\beta}}_1$ and $\hat{\boldsymbol{\beta}}_2$ and the variance-covariance matrix of $\hat{\boldsymbol{\beta}}_1$ can be estimated by $\mathbf{V}_{12} = \mathbf{A}\mathbf{V}_{22}$ and $\mathbf{V}_{11} = -[\partial^2\ell(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2)/\partial\boldsymbol{\beta}_1^2]^{-1} - \mathbf{V}_{12}\mathbf{A}'$, respectively, where

$$\mathbf{A} = \left(\frac{\partial^2\ell(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2)}{\partial\boldsymbol{\beta}_1^2}\right)^{-1}\left(\frac{\partial^2\ell(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2)}{\partial\boldsymbol{\beta}_1\partial\boldsymbol{\beta}_2}\right) .$$

$\mathbf{A}$ is also calculated in the last iteration. If FS-adjustment is used, similar variance estimates can be found, which are based on expected second derivatives. If NRL-adjustment is used, then the variances can be estimated by $(\mathbf{R}_1'\mathbf{R}_1)^{-1}$ in the usual way, whereas for NRL-elimination $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ can be used to estimate $\text{var}(\hat{\boldsymbol{\beta}}_2)$ where $\mathbf{X}$ is defined here to have rows $\mathbf{x}_i'$ from (5.2.7).

5.3 NONLINEAR LEAST SQUARES

In nonlinear least squares, the log-likelihood is

$$\ell(\boldsymbol{\beta},\sigma) = -\tfrac{1}{2}\sigma^{-2}\sum(y_i - \eta_i(\boldsymbol{\beta}))^2 - \log\sigma \ .$$

Since σ^2 can be factorized out, it can be treated as a constant, say $\sigma^2 = 1$, for the purpose of estimating $\boldsymbol{\beta}$. A major application of elimination and adjustment is in nonlinear least squares problems where some parameters occur linearly in $\eta_i(\boldsymbol{\beta})$. We therefore examine models where the vector $\boldsymbol{\eta}$ with elements $\eta_i(\boldsymbol{\beta})$ can be written as

$$\boldsymbol{\eta} = \mathbf{X}_1(\boldsymbol{\beta}_2)\boldsymbol{\beta}_1 \ .$$

For ease of exposition, we restrict our attention to the most common case, which is where each of the p_2 elements of $\boldsymbol{\beta}_2$ is involved in only one of the p_1 columns of $\mathbf{X}_1(\boldsymbol{\beta}_2)$. Results similar to those discussed below also hold for the more general case. Suppose that the i'th element of $\boldsymbol{\beta}_2$, β_{2i}, is used in the c(i)th column of $\mathbf{X}_1(\boldsymbol{\beta}_2)$, $\mathbf{x}_{c(i)}(\boldsymbol{\beta}_2)$. We define the matrix $\mathbf{V}(\boldsymbol{\beta}_2)$ to have ith column $(\partial\mathbf{x}_{c(i)}(\boldsymbol{\beta}_2)/\partial\beta_{2i})$ for $i=1,\ldots,p_2$ and the $p_2\times p_1$ matrix $\mathbf{E}$ to have a one in its $(i,c(i))$th position for $i=1,\ldots,p_2$, and zeros elsewhere. The notation diag($\mathbf{z}$) is used to denote a $p_2\times p_2$ diagonal matrix whose diagonal elements are given by the p_2-vector $\mathbf{z}$.

The NR-elimination/adjustment algorithm is particularly attractive when compared with joint NR here, since

$$\left[\frac{\partial^2\ell(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2)}{\partial\boldsymbol{\beta}_1^2}\right]^{-1} = -\left[\mathbf{X}_1(\boldsymbol{\beta}_2)'\mathbf{X}_1(\boldsymbol{\beta}_2)\right]^{-1},$$

which is needed in Theorem 5.2.1, is also used in the formula $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2) = [\mathbf{X}_1(\boldsymbol{\beta}_2)'\mathbf{X}_1(\boldsymbol{\beta}_2)]^{-1}\mathbf{X}_1(\boldsymbol{\beta}_2)'\mathbf{y}$. The Choleski factorization (5.2.3) is often found during the evaluation of this least squares formula (see Section 2.3). A NR-elimination/adjustment iteration is as fast as joint NR, and convergence is usually better.

The most widely used algorithm for nonlinear least squares is the NRL algorithm which, in the context of nonlinear least squares, is called the Gauss-Newton (GN) algorithm. A joint GN iteration consists of the linear least squares problem of minimizing $S(\beta_1^+,\beta_2^+) = \varepsilon(\beta_1^+,\beta_2^+)'\varepsilon(\beta_1^+,\beta_2^+)$ with respect to β_1^+ and β_2^+, where

$$\varepsilon(\beta_1^+,\beta_2^+) = y - X_1(\beta_2)\beta_1 - X_1(\beta_2)(\beta_1^+-\beta_1) - V(\beta_2)\mathrm{diag}(E\beta_1)(\beta_2^+-\beta_2) \qquad (5.3.1)$$

is a Taylor series for $(y - X_1(\beta_2^+)\beta_1^+)$ around the previous estimate, (β_1,β_2). The matrices X_1 and X_2 in (5.2.4) are equivalent to $X_1(\beta_2)$ and $V(\beta_2).\mathrm{diag}(E\beta_1)$ respectively. Note that the latter is just a rescaling of the columns of $V(\beta_2)$. Joint GN is identical to joint FS.

GN-elimination iterations are based on a similar Taylor series for $(y - X_1(\beta_2^+)\hat{\beta}_1(\beta_2^+))$ around β_2 and can also be expressed as least squares calculations. Lawton and Sylvestre (1971) implemented GN-elimination using numerical derivatives based on differences. Guttman et al (1973) and Golub and Pereyra (1973) derived formulae for the derivatives of $X_1(\beta_2)\hat{\beta}_1(\beta_2)$ and implemented the algorithm using these. Barham and Drane (1972) first used the GN-adjustment method. Kaufman (1975) rediscovered it when attempting to simplify the GN-elimination calculations. Though GN-adjustment is identical to FS-adjustment, GN-elimination is not the same as FS-elimination which, for reasons given in Section 5.2, is impractical.

The joint GN, GN-adjustment and GN-elimination iterations for β_2 can be expressed as

$$\beta_2^+ = \beta_2 + \mathrm{diag}(E\beta_1)^{-1}\,[V(\beta_2)'P_xV(\beta_2)]^{-1}V(\beta_2)'e\ ,$$

$$\beta_2^+ = \beta_2 + \mathrm{diag}(E\hat{\beta}_1(\beta_2))^{-1}\,[V(\beta_2)'P_xV(\beta_2)]^{-1}V(\beta_2)'e\ ,\ \text{and}$$

$$\beta_2^+ = \beta_2 + \mathrm{diag}(E\hat{\beta}_1(\beta_2))^{-1}\,[V(\beta_2)'P_xV(\beta_2) + F(\beta_2)E[X_1(\beta_2)'X_1(\beta_2)]^{-1}E'F(\beta_2)]^{-1}V(\beta_2)'e\ ,$$

respectively, where

$$e = y - X_1(\beta_2)\hat{\beta}_1(\beta_2),$$

$$\mathbf{P}_x = \mathbf{I} - \mathbf{X}_1(\boldsymbol{\beta}_2)[\mathbf{X}_1(\boldsymbol{\beta}_2)'\mathbf{X}_1(\boldsymbol{\beta}_2)]^{-1}\mathbf{X}_1(\boldsymbol{\beta}_2), \text{ and}$$

$$\mathbf{F}(\boldsymbol{\beta}_2) = \text{diag}(\mathbf{V}(\boldsymbol{\beta}_2)'\mathbf{e}).\text{diag}(\mathbf{E}\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2))^{-1}.$$

The calculations are equivalent to least squares regressions of $\mathbf{e}$ against $\mathbf{P}_x\mathbf{V}(\boldsymbol{\beta}_2)$, $\mathbf{P}_x\mathbf{V}(\boldsymbol{\beta}_2)$ and $(\mathbf{P}_x\mathbf{V}(\boldsymbol{\beta}_2) + \mathbf{X}_1(\boldsymbol{\beta}_2)[\mathbf{X}_1(\boldsymbol{\beta}_2)'\mathbf{X}_1(\boldsymbol{\beta}_2)]^{-1}\mathbf{E}'\mathbf{F}(\boldsymbol{\beta}_2))$, respectively. These formulae can be used to compare the algorithms. Since $\mathbf{F}(\boldsymbol{\beta}_2)\mathbf{E}[\mathbf{X}_1(\boldsymbol{\beta}_2)'\mathbf{X}_1(\boldsymbol{\beta}_2)]^{-1}\mathbf{E}'\mathbf{F}(\boldsymbol{\beta}_2)$ is positive semi-definite, the elimination step is usually more conservative than adjustment. If there is only a single parameter in $\boldsymbol{\beta}_2$, the step is always at least as small as the adjustment step. Since $\partial\ell(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)/\partial\boldsymbol{\beta}_2 = \mathbf{0}$ at the least squares estimate, $\mathbf{e}$ is orthogonal to $\mathbf{V}(\boldsymbol{\beta}_2)$, and so $\mathbf{F}(\boldsymbol{\beta}_2)\mathbf{E}[\mathbf{X}_1(\boldsymbol{\beta}_2)'\mathbf{X}_1(\boldsymbol{\beta}_2)]^{-1}\mathbf{E}'\mathbf{F}(\boldsymbol{\beta}_2) = \mathbf{0}$ there. Therefore there is very little difference between GN-adjustment and GN-elimination near the solution. Since the ratio of the GN-adjustment step for β_{2i} to its joint GN step is $\beta_{1,c(i)}/\hat{\beta}_{1,c(i)}(\boldsymbol{\beta}_2)$, there is also very little difference between the iterations for $\boldsymbol{\beta}_2$ in these algorithms when they approach the least squares estimates. Therefore all three GN algorithms have the same asymptotic convergence. Ruhe and Wedin (1980) describe related theoretical results about asymptotic convergence rates.

The main differences between the behaviour of the GN algorithms are in the initial iterations. Kaufman (1975) showed in numerical examples that GN-elimination and GN-adjustment took similar numbers of iterations and, as discussed in Section 5.2 above, GN-adjustment usually takes fewer iterations than joint GN.

We next compare the speed per iteration of the GN algorithms. The adjustment and elimination implementations described below are similar to the numerically sound methods described by Golub and Pereyra (1973) and Kaufman (1975), and a standard QR algorithm is used for joint GN. Details of column interchanges that may be required if $\mathbf{X}_1(\boldsymbol{\beta}_2)$ is singular are omitted for simplicity. First, an orthogonal matrix $\mathbf{Q}$ is found such that

$$\mathbf{Q}\,[\mathbf{X}_1(\boldsymbol{\beta}_2) : \mathbf{V}(\boldsymbol{\beta}_2) : \mathbf{y}] = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{S}_1 & \mathbf{r}_{21} \\ \mathbf{0} & \mathbf{S}_2 & \mathbf{w} \end{bmatrix} \qquad (5.3.2)$$

where $\mathbf{R}_{11}$ is upper triangular. $\mathbf{R}_{11}$ and $\mathbf{r}_{21}$ are as in (5.2.4) and the comumns of $\mathbf{S}_1$ are proportional to those of $\mathbf{R}_{21}$. $\mathbf{Q}$ can be a product of Householder matrices or a product of Givens matrices and does not need to be stored. In the joint GN and GN-adjustment algorithms, $\mathrm{diag}(\mathbf{E}\boldsymbol{\beta}_1)(\boldsymbol{\beta}_2^+-\boldsymbol{\beta}_2)$ and $\mathrm{diag}(\mathbf{E}\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2))(\boldsymbol{\beta}_2^+-\boldsymbol{\beta}_2)$ are each given by the regression coefficients of $\mathbf{S}_2$ against $\mathbf{w}$ (found using a QR algorithm). For GN-elimination, $\mathrm{diag}(\mathbf{E}\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2))(\boldsymbol{\beta}_2^+-\boldsymbol{\beta}_2)$ is the vector of regression coefficients of

$$\begin{bmatrix} \mathbf{S}_1^* \\ \mathbf{S}_2 \end{bmatrix} \quad \text{against} \quad \begin{bmatrix} \mathbf{0} \\ \mathbf{w} \end{bmatrix}$$

where $\mathbf{S}_1^* = \mathbf{P}.\mathrm{diag}(\mathbf{S}_2'\mathbf{w})\mathrm{diag}(\mathbf{E}\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2))^{-1}$ and the columns of $\mathbf{P}$ are found by forward substitution from $\mathbf{R}_{11}'\mathbf{P}=\mathbf{E}'$. In the adjustment and elimination algorithms, $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ is found by solving $\mathbf{R}_{11}\boldsymbol{\beta}_1 = \mathbf{r}_{21}$ by back-substitution and has residual sum of squares $\mathbf{w}'\mathbf{w}$. In the joint algorithm, $\boldsymbol{\beta}_1^+$ is found by solving $\mathbf{R}_{11}\boldsymbol{\beta}_1^+ = \mathbf{r}_{21}-\mathbf{S}_1\mathrm{diag}(\mathbf{E}\boldsymbol{\beta}_1)(\boldsymbol{\beta}_2^+-\boldsymbol{\beta}_2)$ and the residual sum of squares must be evaluated explicitly. Therefore the adjustment algorithm is faster than the others, although the differences are usually slight. Golub and Pereyra (1973) describe the implementation of a Levenberg-Marquardt stepsize modification that ensures convergence for all algorithms. As discussed in Section 5.2, this usually takes longer for GN-elimination and GN-adjustment than for joint GN.

It should be noted that in the GN-elimination and GN-adjustment algorithms, the computation above consists of the end of one iteration (evaluation of $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$) and the start of the next iteration (evaluation of $\boldsymbol{\beta}_2^+$). If the need for a stepsize modification is detected after $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ is evaluated, the stepsize previously taken to evaluate $\boldsymbol{\beta}_2$ must be modified. It is therefore more efficient to perform the computations as described above rather than initially evaluating $\mathbf{R}$ in (5.2.4) and basing the computations on $\mathbf{R}$ since arranging the computations as above means that no unnecessary calculations are

performed if a stepsize modification is needed and therefore (5.3.2) must be recalculated.

The algorithm as described above however needs much more storage than algorithms that use Givens rotations to initially reduce $[\mathbf{X}_1(\boldsymbol{\beta}_2) : \mathbf{V}(\boldsymbol{\beta}_2) : \mathbf{y}]$ to upper triangular form. Since this is as fast for iterations in which no stepsize modification is needed, it would often be preferred, despite its relative inefficiency in iterations in which stepsize modifications are needed.

As noted by Harville (1973), often some columns of $\mathbf{X}_1(\boldsymbol{\beta}_2)$ do not involve $\boldsymbol{\beta}_2$, say $\mathbf{X}_1(\boldsymbol{\beta}_2) = [\mathbf{X}_{11} : \mathbf{X}_{12}(\boldsymbol{\beta}_2)]$. All GN algorithms can make use of this fact to increase the speed per iteration. Since the initial Householder transformations to reduce $\mathbf{X}_{11}$ to upper triangular are the same in each iteration, these transformations can be saved and after the first iteration they only need to be applied to $\mathbf{X}_{12}(\boldsymbol{\beta}_2)$ and $\mathbf{V}(\boldsymbol{\beta}_2)$, but not $\mathbf{X}_{11}$.

GN-adjustment would usually be preferred to joint GN. Since it is also simpler than GN-elimination, it is probably the most satisfactory GN algorithm. Unfortunately, comparisons between the NR and GN algorithms cannot be made so easily. The NR algorithms, in general, seem to need more corrective action than GN in the first few iterations to get convergence; but once they are near enough to the solution, the NR methods converge faster. The NR methods usually need longer and more complex programs, but can often make use of any sparseness existing in $\mathbf{X}_1(\boldsymbol{\beta}_2)$ and can be faster in execution time per iteration, if the second derivatives of $\mathbf{X}_1(\boldsymbol{\beta}_2)$ are not too difficult to evaluate. It would be expected that the GN algorithms would determine the optimum $\boldsymbol{\beta}$ more accurately than the NR algorithms, since numerically accurate QR methods are used in each iteration (this is partly why the GN iterations are often slower). Finally it is noted that Dennis et al (1981) proposed a new nonlinear least squares algorithm which has some of the benefits of both NR and GN; the application of elimination and adjustment to their algorithm is an area for further study.

5.4 A NONLINEAR LEAST SQUARES EXAMPLE

In this section the nonlinear least squares algorithms described in Section 5.3 are compared utilizing a numerical example which arises in bioassay. The data in Table 5.4.1, which were published by Ross (1970), are assumed to satisfy the model.

$$\begin{aligned} &\text{Control group} &&: E[y] = \alpha_5 \\ &\text{Substance 1} &&: E[y] = \alpha_5 + \alpha_6 F(\alpha_1 + \alpha_2 x) \\ &\text{Substance 2} &&: E[y] = \alpha_5 + \alpha_7 F(\alpha_3 + \alpha_4 x) \end{aligned} \qquad (5.4.1)$$

where $F(z) = \exp(z)/(1+\exp(z))$ is the sigmoid logistic curve, x is the log dose of the substance and the response y is assumed to be normally distributed with constant variance.

TABLE 5.4.1
Data from Ross (1970)

	Log Dose, x	Response, y
Control	--	87.08
Substance 1	1.59934	98.60
	1.90940	109.22
	2.07733	127.07
	2.31160	145.27
	2.52957	161.83
Substance 2	1.36398	91.13
	1.91840	111.57
	2.08123	114.75
	2.32533	130.68
	2.56949	128.48

To get initial estimates, the asymptotes of the response curves were first estimated by eye from plots of y against x to give $\beta_1' = [\alpha_5\ \alpha_6\ \alpha_7] = [80\ 120\ 70]$. Next $F^{-1}((y-\alpha_5)/\alpha_6)$ was plotted against x

for substance 1 and a straight line was fitted by eye. From that and a corresponding plot for substance 2, initial estimates for the nonlinear parameters were taken as $\boldsymbol{\beta}_2' = [\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4] = [-6.29\ 2.78\ -4.94\ 2.38]$.

The five algorithms used to find the least squares estimate of $\boldsymbol{\beta}' = [\boldsymbol{\beta}_1\ \boldsymbol{\beta}_2]$ were joint NR, NR elimination/adjustment, joint GN, GN adjustment and GN elimination. In all cases the sparseness of $\mathbf{I}(\boldsymbol{\beta}_2)$ was used to reduce the number of floating point operations. Both NR algorithms used Choleski factorizations, whereas the GN algorithms used Householder transformations.

TABLE 5.4.2

Residual Sum of Squares After Each Iteration of Five Algorithms for Fitting Model (5.4.1) to the Data in Table 5.4.1 by Least Squares.

Iteration	Algorithm				
	Joint NR	NR Elimination/ Adjustment	Joint GN	GN Adjustment	GN Elimination
0	139.9236274	106.0399378	139.9236274	106.0399378	106.0399378
1	120.3887294	100.9095236	75.3712635	69.1800370	68.3619778
2	74.6159456	54.7496706	68.6379077	53.4596291	53.4169685
3	55.6485499	52.9890895	53.2769362	52.9490063	52.9456172
4	53.0267833	52.9212025	52.9344197	52.9227832	52.9224970
5	52.9212045	52.9210154	52.9218627	52.9211471	52.9211255
6	52.9210154	52.9210154	52.9210828	52.9210259	52.9210242
7	52.9210154		52.9210208	52.9210163	52.9210161
8			52.9210159	52.9210155	52.9210155
9			52.9210155	52.9210155	52.9210155
10			52.9210154	52.9210154	52.9210154
11			52.9210154	52.9210154	52.9210154

Table 5.4.2 shows the convergence of the residual sum of squares for all algorithms. The second derivative matrix stayed positive definite for both NR algorithms, but each needed a line search in one iteration to get a reduction in the residual sum of squares (the stepsize algorithm of Bard (1974, pages 119-113) was used). Stepsize

modifications were not needed for any of the GN algorithms. All algorithms successfully converged to the least squares estimate of β' = [87.7383 95.0949 44.9589 -8.24176 3.75679 -8.73717 4.54506]. The GN algorithms were all similar, with joint GN about one iteration behind the other methods. Once the NR algorithms got close to the solution, they both rapidly converged with NR-elimination/adjustment taking fewer iterations.

TABLE 5.4.3

Number of Floating Point Operations per Iteration of Five Algorithms for Fitting Model (5.4.1) to the Data in Table 5.4.1. The Additional Operations that would be Needed for an Extra Observation with Either Substance are Given in Brackets.

Operation	Algorithm				
	Joint NR	NR Elimination/ Adjustment	Joint GN	GN Adjustment	GN Elimination
*,/	347(22)	332(23)	831(76)	817(75)	917(77)
+,-	336(23)	333(25)	737(69)	702(67)	777(69)
Square root, Exponential	10(1)	10(1)	24(1)	24(1)	24(1)

The numbers of floating point operations per iteration were also recorded and are shown in Table 5.4.3; the figures in brackets are the extra number of operations per iteration that would be required for each additional observation getting substance 1 or 2. The NR iterations were considerably faster than GN, but there was little variation within each grouping. It was expected that the NR algorithms would determine the least squares solutions relatively inaccurately, compared with GN, but no differences in accuracy were noticed between

32-bit single precision versions, when they were compared to the double precision solutions.

5.5 OTHER APPLICATIONS OF ELIMINATION AND ADJUSTMENT

In this section, we describe some applications of elimination and adjustment to models other than least squares. As pointed out by Ross (1982), elimination can be used in models where $y_1,\ldots,y_n$ are independent observations from Poisson or gamma distributions with means $\mu_i = \beta_1 f_i(\boldsymbol{\beta}_2)$. Then the scalar β_1 can be eliminated as a weighted mean of $y_1,\ldots,y_n$. Note that $\boldsymbol{\beta}_2$ is usually the parameter of interest and the elimination and adjustment algorithms immediately provide the variance of $\hat{\boldsymbol{\beta}}_2$. A similar type of elimination and adjustment can be used in some heteroscedastic normal models from Section 4.3. For example, in models where the variance is specified by (4.3.2), α can be eliminated as

$$\hat{\alpha} = \log\left(n^{-1} \sum \frac{(y_i - \mu_i)^2}{\exp(\gamma.\log(\mu_i))} \right) .$$

The NRL-adjustment and FS-adjustment algorithms can therefore be implemented easily by adjusting α after each iteration of joint NRL or joint FS with

$$\alpha^{++} = \alpha + \log\left(n^{-1} \sum \frac{(y_i - \mu_i)^2}{\sigma_i^2} \right) .$$

The summation on the right is required for evaluating the log-likelihood after each iteration and the adjustment algorithms are therefore as fast as their unadjusted versions. For example, NRL-adjustment and FS-adjustment were applied to the stopping distance example in Figure 4.3.1. The iterations are given in Table 5.5.1 and both algorithms are slightly better than their unadjusted versions (Table 4.3.5).

In normal models where $E[y_i] = \mathbf{x}_i'\boldsymbol{\beta}_1$ and $\text{var}(y_i) = v(\boldsymbol{\beta}_2)$, $\hat{\boldsymbol{\beta}}_1$ can be expressed as a function of $\boldsymbol{\beta}_2$ with a standard weighted least squares calculation. FS-adjustment is identical to the FS-alternation

TABLE 5.5.1

FS- and NRL-adjustment Algorithms Applied to Stopping Distance Example

Iteration	β_1	β_2	α	γ	Log-Likelihood	Convergence Rate, C
FS-adjustment Algorithm						
0	0.55526	0.06269	4.53674	0.00000	-174.40727	
1	0.55526	0.06269	1.25511	0.84734	-158.48830	0.37782
2	0.61396	0.06050	-0.21765	1.26210	-155.53105	0.22290
3	0.64006	0.05935	-0.56076	1.36271	-155.37635	0.15732
4	0.64650	0.05905	-0.61301	1.37809	-155.37222	0.15406
5	0.64741	0.05900	-0.62117	1.38051	-155.37212	0.14869
6	0.64755	0.05899	-0.62238	1.38087	-155.37212	0.15115
7	0.64757	0.05899	-0.62256	1.38092	-155.37212	0.14933
8	0.64758	0.05899	-0.62259	1.38093	-155.37212	0.15053
NRL-adjustment Algorithm						
0	0.55526	0.06269	4.53674	0.00000	-174.40727	
1	0.73634	0.05644	-3.43932	2.31495	-165.87150	0.57895
2	0.68539	0.05570	-0.42260	1.32734	-155.71696	0.07670
3	0.64739	0.05892	-0.60404	1.37567	-155.37298	0.08167
4	0.64759	0.05899	-0.62260	1.38094	-155.37212	0.00126
5	0.64758	0.05899	-0.62259	1.38093	-155.37212	0.00323

algorithm described in Section 4.4. The NRL algorithm is identical to NR, and NRL-adjustment is therefore identical to NRL-elimination. To implement these efficiently, note that the calculations to evaluate $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ immediately give $\mathbf{R}_{11}$ in (5.2.4). However the remaining matrices in (5.2.4) cannot be easily found using least squares calculations by the method in (4.1.1) without reevaluating $\mathbf{R}_{11}$. If there are only one or two components in $\boldsymbol{\beta}_2$, then the partial derivatives with respect to $\boldsymbol{\beta}_2$ can be evaluated and incorporated in $\mathbf{R}_{11}$ to form $\mathbf{R}$ in (5.2.4) using subroutine NUISNC in Appendix B. If the calculations are done in this way, an iteration of NRL-elimination/adjustment is as fast as one of joint NRL.

In some models where no single parameter can be eliminated or adjusted, a group of parameters can be simultaneously adjusted. For example, in the Poisson model with $E[y_i] = \mathbf{x}_i'\boldsymbol{\beta}$, no parameter can be explicitly eliminated. However if the model is rewritten in the

equivalent form $E[y_i] = \alpha(\mathbf{x}_i'\boldsymbol{\beta})$, then

$$\hat{\alpha} = \frac{\sum y_i}{\sum (\mathbf{x}_i'\boldsymbol{\beta})} .$$

Therefore it is possible to adjust $\boldsymbol{\beta}$ after a NR or FS iteration with

$$\boldsymbol{\beta}^{++} = \boldsymbol{\beta}^{+} \times \left(\frac{\sum y_i}{\sum (\mathbf{x}_i'\boldsymbol{\beta}^{+})} \right) .$$

As both summations on the right are required to evaluate the log-likelihood, the adjustment algorithms are as fast as the unadjusted algorithms. Unfortunately, adjustment has no effect on the FS algorithm since the "responses" are not affected and the weights are all changed by the same proportion. However, adjustment does have an effect on the NR iterations. For example, NR-adjustment was applied to data sets 1 and 2 in Table 3.4.1, and its iterations are shown in Table 5.5.2. When compared with Tables 3.4.2 and 3.4.3, it can be seen that adjustment offers an improvement for data set 1; however, for data set 2 NR-adjustment needs stepsize modifications in the initial iterations and is slower than joint NR. Adjustment and elimination do not <u>always</u> result in fewer iterations.

In a similar way, for normal models with $E[y_i] = \mu_i$ and $\text{var}(y_i) = \mathbf{x}_i'\boldsymbol{\beta}_2$, $\boldsymbol{\beta}_2$ can be adjusted with

$$\boldsymbol{\beta}_2^{++} = \boldsymbol{\beta}_2^{+} \cdot n^{-1} \sum \frac{(y_i-\mu_i)^2}{\mathbf{x}_i'\boldsymbol{\beta}_2^{+}}$$

Adjustment is even possible when $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ cannot be explicitly found. For example, if a model has two systematic parts $\eta_i^{(1)}(\boldsymbol{\beta}_1)$ and $\eta_i^{(2)}(\boldsymbol{\beta}_2)$, then $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$ can be numerically found for fixed $\boldsymbol{\beta}_2$ with the IRLS calculations of Chapter 3. Theorem 5.2.1 can then be used for NR-adjustment iterations on $\boldsymbol{\beta}_2$ and NRL- and FS-adjustment iterations can be implemented in a similar way. (Note that $[\partial^2\ell/\partial\boldsymbol{\beta}_1{}^2]^{-1}$ (or $E[\partial^2\ell/\partial\boldsymbol{\beta}_1{}^2]^{-1}$) is found in the last IRLS iteration). These algorithms have the advantage of simplicity over the corresponding joint algorithm since definiteness and stepsize modifications are less often needed, but they usually take longer to converge since each iteration may involve several IRLS sub-iterations to find $\hat{\boldsymbol{\beta}}_1(\boldsymbol{\beta}_2)$. There does not

TABLE 5.5.2

Iterations of NR-adjustment Applied to Data Sets 1 and 2 in Table 3.4.1 Starting Values (Iteration 0) were Found as Decribed in the Text.

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C
Data Set 1				
0	4.84790	-0.08263	-6.98886	
1	8.89957	-1.43319	-6.36573	0.26532
2	8.18230	-1.19410	-6.30154	0.15573
3	8.05275	-1.15092	-6.30008	0.02078
4	8.05000	-1.15000	-6.30008	0.00042
5	8.05000	-1.15000	-6.30008	0.00000
Data Set 2				
0	1.67619	0.90794	-11.69004	
1	(12.42552	-2.67517	-∞)	
	3.66344	0.24552	-10.07241	0.78197
2	(13.28434	-2.96145	-∞)	
	5.62696	-0.40899	-8.92801	0.72450
3	(12.51020	-2.70340	-∞)	
	7.17134	-0.92378	-8.28336	0.70091
4	(11.75022	-2.45007	-∞)	
	8.25688	-1.28563	-7.95585	0.70007
5	(11.29676	-2.29892	-∞)	
	8.99738	-1.53246	-7.79275	0.70775
6	(11.05624	-2.21875	-∞)	
	9.50545	-1.70182	-7.71040	0.71668
7	10.93135	-2.17712	-7.62045	0.10949
8	10.79247	-2.13082	-7.61926	0.01306
9	10.79064	-2.13021	-7.61926	0.00017
10	10.79064	-2.13021	-7.61926	0.00000

seem to be much to be gained from using adjustment or elimination in this way.

As a final example where numerical elimination can be helpful, consider p populations where the ith (i=1,...,p) has observations y_{ij} and log-likelihood components of the form $\ell_{ij}(\alpha_i, \gamma_i)$ for $j=1,\ldots,n_i$. To obtain parameter estimates with common $\alpha_i = \alpha$, we can consider elimination of α by $\hat{\alpha}(\gamma_1,\ldots,\gamma_p)$ which can be numerically evaluated by a unidimensional optimization. A NR elimination step for $\gamma_1,\ldots,\gamma_p$ based on Theorem 5.2.1 can then be shown to be

$$\gamma_i^+ = \gamma_i - d_i^{-1} \left\{ b_i + a_i \left(\frac{\Sigma d_j^{-1} a_j b_j}{C - \Sigma d_j^{-1} a_j^2} \right) \right\} \qquad (5.5.1)$$

where

$$b_i = \sum_j \partial \ell_{ij}(\alpha, \gamma_i)/\partial \gamma_i \ ,$$

$$d_i = \sum_j \partial^2 \ell_{ij}(\alpha, \gamma_i)/\partial \gamma_i^2 \ ,$$

$$a_i = \sum_j \partial^2 \ell_{ij}(\alpha, \gamma_i)/\partial \alpha \partial \gamma_i \ , \text{ and}$$

$$C = \sum_i \sum_j \partial^2 \ell_{ij}(\alpha, \gamma_i)/\partial \alpha^2$$

for $i = 1, \ldots, p$.

White and Eberhardt (1980) considered fitting negative binomial models to elk pellet-group counts which were collected from various sites at $p = 5$ different times. They were able to interpret the four models which had (a) different mean and shape parameter at each time, (b) different means, common shape, (c) common mean, different shape, and (d) common mean and shape. In all models other than (c), the means can be explicitly eliminated and replaced by either the population or overall mean, reducing the problem to either 1 or p univariate optimizations. In model (c) the common mean can be eliminated numerically and equation (5.5.1) can then be used to update the shape parameters.

5.6 SYSTEMATIC PARTS WITH LINEAR AND NONLINEAR PARAMETERS

In this section, a technique is described for improving NRL iterations for models with a systematic part which has both linear and nonlinear parameters. This is closely related to the GN-adjustment algorithm for nonlinear least squares.

We first consider a general model with a single systematic part and log-likelihood components $\ell_i(\eta_i)$. If we denote the Taylor series approximation of $\ell_i(\eta_i^+)$ round the value η_i by $\ell_i^*(\eta_i^+)$, then

$$\ell_i^*(\eta_i^+) = \ell_i(\eta_i) + (\eta_i^+ - \eta_i)\,\ell_i'(\eta_i) + (\eta_i^+ - \eta_i)^2 \ell_i''(\eta_i) / 2$$

Maximizing $\sum \ell_i^*(\eta_i^+)$ with respect to $\boldsymbol{\beta}^+$ is therefore identical to maximizing the sum of squares,

$$\ell^*(\boldsymbol{\beta}^+) = \frac{1}{2} \sum \left(\eta_i + \frac{\ell_i'(\eta_i)}{\ell_i''(\eta_i)} - \eta_i(\boldsymbol{\beta}^+)\right)^2 \ell_i''(\eta_i) \qquad (5.6.1)$$

with respect to $\boldsymbol{\beta}^+$. One iteration of the GN algorithm applied to this nonlinear least squares problem is identical to an iteration of the NRL algorithm applied to $\ell(\boldsymbol{\beta})$. Similarly, if the systematic part η_i is non-random an iteration of GN applied to the similar approximation with $\ell_i''(\eta_i)$ replaced by its expected value is identical to the FS algorithm applied to $\ell(\boldsymbol{\beta})$. In models where the systematic part has a structure like that considered in Section 5.3, with $\boldsymbol{\eta} = \mathbf{X}_1(\boldsymbol{\beta}_2)\boldsymbol{\beta}_1$, it is therefore tempting to try to improve on the NRL algorithm by using an iteration of GN-adjustment for this sub-problem. Unfortunately however, as discussed in Section 5.3, a single GN-adjustment iteration spans the calculations that would be involved in <u>two</u> joint GN iterations, the adjustment being performed as the initial part of the second joint GN iteration. A single GN-adjustment iteration therefore takes considerably longer than a single joint GN iteration and it is unlikely that the benefits in terms of improved convergence would outweigh this increased execution time per iteration.

However a single joint GN iteration can be improved slightly with little overhead. A QR algorithm can be applied to $[\mathbf{X}_1(\boldsymbol{\beta}_2) : \mathbf{V}(\boldsymbol{\beta}_2) : \mathbf{z}]$ with weights given by $\{\ell_i'(\eta_i)\}$ where $\mathbf{V}(\boldsymbol{\beta}_2)$ is as defined in Section 5.3 and $\mathbf{z}$ has i'th element $z_i = \ell_i'(\eta_i)/\ell_i'(\eta_i)$, to give upper triangular matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{S}_1 & \mathbf{r}_{21} \\ \mathbf{0} & \mathbf{S}_2^* & \mathbf{r}_{22} \\ \mathbf{0} & \mathbf{0} & \mathbf{r}_3 \end{bmatrix}$$

Then the joint GN algorithm defines $\boldsymbol{\beta}_1^+$ and $\boldsymbol{\beta}_2^+$ by the equations $\mathbf{S}_2^*\text{diag}(\mathbf{E}\boldsymbol{\beta}_1).(\boldsymbol{\beta}_2^+-\boldsymbol{\beta}_2) = \mathbf{r}_{22}$ and $\mathbf{R}_{11}(\boldsymbol{\beta}_1^+-\boldsymbol{\beta}_1) = \mathbf{r}_{21}-\mathbf{S}_1\mathbf{S}_2^{*-1}\mathbf{r}_{22}$. However, if $\boldsymbol{\beta}_1$ was set at $\boldsymbol{\beta}_1^+$ before this joint GN iteration on the nonlinear least squares sub-problem, the iteration for $\boldsymbol{\beta}_2$ would have been defined by $\mathbf{S}_2^*\text{diag}(\mathbf{E}\boldsymbol{\beta}_1^+).(\boldsymbol{\beta}_2^{++}-\boldsymbol{\beta}_2) = \mathbf{r}_{22}$ and this would be expected to be an improvement over $\boldsymbol{\beta}_2^+$. This improvement can be implemented after a joint NRL iteration by multiplying $\boldsymbol{\beta}_2^+$ by $\text{diag}(\mathbf{E}\boldsymbol{\beta}_1^+).\text{diag}(\mathbf{E}\boldsymbol{\beta}_1)^{-1}$, which can be done with little overhead on the execution time for the iteration. We call the technique the NRL algorithm with improvement of $\boldsymbol{\beta}_2$; improvement of $\boldsymbol{\beta}_2$ can be similarly applied to the FS algorithm.

As an application of this algorithm, it was applied to the probit residual responses model defined by (3.5.4). The improvement for $\boldsymbol{\beta}_2$ after iterations of the joint NRL algorithm (A7) and the FS algorithm (A6) can be written as

$$(\boldsymbol{\beta}_2^{++} - \boldsymbol{\beta}_2) = \frac{1-\beta_1^+}{1-\beta_1}(\boldsymbol{\beta}_2^+ - \boldsymbol{\beta}_2)$$

The algorithms were applied to the data in Table 3.5.7 and their iterations are shown in Table 5.6.1. The algorithms with improvement of $\boldsymbol{\beta}_2$ perform considerably better than ordinary NRL and FS, whose iterations were shown in Table 3.5.8.

TABLE 5.6.1

Iterations of the NRL and FS Algorithms with Improvement of β_2 and β_3 Applied to Fit Residual Responses Model to Data in Table 3.5.7.

Iteration	β_1	β_2	β_3	Log-Likelihood	Convergence Rate, C
FS Algorithm with Improvement					
0	0.14546	-2.29687	0.74910	-1.46750	
1	0.25874	-3.32866	0.91608	-1.23498	0.45081
2	0.22160	-4.35373	1.19934	-0.92499	0.36319
3	0.20553	-3.97638	1.11683	-0.91902	0.30952
4	0.21568	-4.07876	1.13044	-0.91751	0.13842
5	0.21127	-4.06851	1.13298	-0.91701	0.22552
6	0.21247	-4.06627	1.13076	-0.91695	0.61005
7	0.21223	-4.06813	1.13159	-0.91694	0.36354
8	0.21225	-4.06740	1.13135	-0.91694	0.27739
9	0.21226	-4.06760	1.13140	-0.91694	0.22067
NRL Algorithm(A7) with Improvement					
0	0.14434	-2.29175	0.74842	-1.47056	
1	0.21820	-3.29438	0.92522	-1.04787	0.44249
2	0.22761	-4.16397	1.13736	-0.92743	0.11948
3	0.20773	-4.04185	1.13195	-0.91824	0.26158
4	0.21347	-4.07316	1.13131	-0.91702	0.22367
5	0.21198	-4.06721	1.13170	-0.91695	0.13662
6	0.21231	-4.06761	1.13134	-0.91694	0.16346
7	0.21225	-4.06760	1.13142	-0.91694	0.41168
8	0.21226	-4.06756	1.13139	-0.91694	0.24592

5.7 CONCLUDING REMARKS ABOUT ELIMINATION AND ADJUSTMENT

With all types of algorithm, adjustment of β_1 to $\hat{\beta}_1(\beta_2)$ between joint iterations for β_1 and β_2 increases the likelihood. Convergence would be expected with fewer iterations than when using joint iterations without adjustment. Also fewer starting values are required, modifications to the basic algorithm to ensure convergence are usually needed less often, and these modifications are easier to apply.

In nonlinear least squares an iteration of the adjustment versions of NR and GN is as fast as an iteration of the corresponding joint algorithm provided a stepsize modification is not needed. However if a full adjustment step reduces the likelihood, extra evaluations of $\hat{\beta}_1(\beta_2)$ will be needed to reevaluate the likelihood with reduced stepsize, so that such step reductions are slower for adjustment iterations than for joint iterations. Several other types of model were identified where adjustment can be done with little overhead on joint NR, NRL or FS iterations.

Adjustment does not always give faster convergence than the corresponding joint algorithm. Published numerical results for various nonlinear least squares problems vary from joint algorithms being marginally faster than adjustment algorithms to the adjustment algorithms being considerably faster. In view of the considerable potential gain and slight possible loss, adjustment would therefore seem generally advisable, at least for nonlinear least squares problems. In other models where there is no overhead to evaluating $\hat{\beta}_1(\beta_2)$, such as in some of the examples in Section 3.5, adjustment is also helpful.

Elimination is identical to adjustment when applied to the NR algorithm, FS elimination cannot usually be implemented, and GN

elimination is more complex than FS adjustment with existing published results no faster. Therefore there is currently no reason to recommend elimination rather than adjustment.

In models with a systematic part that has linear and non-linear parameters, the NRL and FS algorithms are often speeded by applying a related type of improvement after each iteration, as described in Section 5.6.

6. THE EM ALGORITHM

6.1 GENERAL DESCRIPTION OF THE EM ALGORITHM

In this thesis, we are largely concerned with general algorithms that can be implemented using a sequence of weighted least squares calculations (IRLS algorithms). The algorithms described in earlier chapters have all been based on the Newton-Raphson (NR) algorithm or modifications of it such as Fisher's scoring technique (FS) and the NRL algorithm. In this chapter, we consider a different type of algorithm called the EM algorithm. Strictly speaking, the EM algorithm is a method of transforming the problem of maximizing a likelihood function into a sequence of maximizations of related functions. These sub-maximizations can be performed using any standard algorithm such as NR, FS or NRL, but the benefits of the method are usually found in problems where there are explicit solutions to the sub-problems. In some applications, the sub-maximizations can be performed using least squares calculations.

In this section, we define the EM algorithm and examine its properties. In later sections, we consider various application areas of the algorithm; for each, we examine whether the algorithm can be implemented with IRLS and make some comparisons with the NR, FS and NRL algorithms.

The EM algorithm has been used for many years in particular types of model such as in experimental designs that have been unbalanced by missing observations. It was first described in a general form by Dempster et al (1977) for models in which the probability density function of a vector of observed data $\mathbf{y}$ can be written in the form

$$f(\mathbf{y} \mid \boldsymbol{\beta}) = \int_{\mathbf{y}_c \in Y_c(\mathbf{y})} f_c(\mathbf{y}_c \mid \boldsymbol{\beta})\, d\mathbf{y}_c \qquad (6.1.1)$$

where $Y_c(\mathbf{y})$ is some subset of the values of $\mathbf{y}_c$ and $f_c(\mathbf{y}_c|\boldsymbol{\beta}) \geq 0$ for all $\mathbf{y}_c$ and $\boldsymbol{\beta}$. Models can be most readily written in this way if the distribution of $\mathbf{y}$ can be generated from some random vector $\mathbf{y}_c$ which can be thought of as the "complete" data and if $\mathbf{y}=\mathbf{y}(\mathbf{y}_c)$ can be treated as observed "incomplete" data. In some applications, the "complete" data may have some physical meaning; for example, the "incomplete" data may be recorded because of missing observations or grouping in some "complete" data. In other applications, the "complete" data does not correspond to any aspect of the real system being modelled, and is just a conceptual device to express the distribution in the required form. This will become clearer when applications of the algorithm are discussed in later sections.

We shall extend the class of models to which the EM algorithm can be applied by relaxing slightly the requirement of Dempster et al (1977) that $f_c(\mathbf{y}_c|\boldsymbol{\beta}) \geq 0$ for all $\mathbf{y}_c$ and $\boldsymbol{\beta}$. We shall instead treat all models for which $f_c(\mathbf{y}_c|\boldsymbol{\beta}).k(\mathbf{y}_c) \geq 0$ for all $\mathbf{y}_c$ and $\boldsymbol{\beta}$ and some function $k(\cdot)$. A class of models of this more general type will be used in Section 6.6.

In its most general form, each iteration of the EM algorithm can be expressed as

maximize with respect to $\boldsymbol{\beta}^+$ the function

$$\int \log \{ f_c(\mathbf{y}_c|\boldsymbol{\beta}^+).k(\mathbf{y}_c) \} \cdot \frac{f_c(\mathbf{y}_c|\boldsymbol{\beta})}{f(\mathbf{y}|\boldsymbol{\beta})}\, d\mathbf{y}_c \qquad (6.1.2)$$

where $\boldsymbol{\beta}$ is the value from the previous iteration. When $\mathbf{y}_c$ and $\mathbf{y}$ can be interpreted as complete and incomplete data, (6.1.2) can be written as

maximize with respect to $\boldsymbol{\beta}^+$ the function

$$E[\, \log f_c(\mathbf{y}_c|\boldsymbol{\beta}^+) \mid \mathbf{y}, \boldsymbol{\beta} \,] \qquad (6.1.3)$$

Each EM iteration therefore involves an optimization of the same size as the original optimization problem. Clearly, there cannot be any

gain unless (6.1.3) is considerably simpler than the original optimization of $\log f(\mathbf{y}|\boldsymbol{\beta})$. The main case where this happens is when the complete density is in the exponential family

$$f_c(\mathbf{y}_c|\boldsymbol{\beta}) = b(\mathbf{y}_c).\exp(\mathbf{t}(\mathbf{y}_c)'\boldsymbol{\beta}) \;/\; a(\boldsymbol{\beta}) \qquad (6.1.4)$$

so that (6.1.3) is equivalent to

maximize with respect to $\boldsymbol{\beta}^+$ the function

$$E[\, \mathbf{t}(\mathbf{y}_c) \mid \mathbf{y}\,,\, \boldsymbol{\beta}\,]'\boldsymbol{\beta}^+ - \log a(\boldsymbol{\beta}^+) \qquad (6.1.5)$$

Here each iteration of the algorithm consists of two steps,

E step : Evaluate $\mathbf{t}^*(\mathbf{y},\boldsymbol{\beta}) = E[\, \mathbf{t}(\mathbf{y}_c) \mid \mathbf{y},\boldsymbol{\beta}]$

M step : Maximize with respect to $\boldsymbol{\beta}^+$ the function

$$\mathbf{t}^*(\mathbf{y},\boldsymbol{\beta})'\boldsymbol{\beta}^+ - \log a(\boldsymbol{\beta}^+) \qquad (6.1.6)$$

which gives rise to the algorithm's name. In practice, the algorithm is mainly used for problems where the "complete" density can be written in the form (6.1.4).

We shall now prove convergence of the general algorithm. If the algorithm converges to some $\hat{\boldsymbol{\beta}}$ and if the maximization in the final iteration is to a turning point, then

$$\left[\frac{\partial}{\partial \boldsymbol{\beta}^+} \int_{\mathbf{y}_c \in Y_c(\mathbf{y})} \log \{f_c(\mathbf{y}_c|\boldsymbol{\beta}^+).k(\mathbf{y}_c)\}.f_c(\mathbf{y}_c|\boldsymbol{\beta})\, d\mathbf{y}_c \right]_{\boldsymbol{\beta}^+=\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} = \mathbf{0}$$

$$\text{i.e.} \quad \left[\int \frac{\partial f_c(\mathbf{y}_c|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\, d\mathbf{y}_c \right]_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} = \mathbf{0}$$

$$\text{i.e.} \quad \left[\frac{\partial\, f(\mathbf{y}|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \right]_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} = \mathbf{0}$$

so that $\hat{\boldsymbol{\beta}}$ is a local maximum of the likelihood. Dempster et al(1977) also show that if $\boldsymbol{\beta}^+$ maximizes (6.1.2), then

$$\log f(\mathbf{y}|\boldsymbol{\beta}^+) = \log \int f_c(\mathbf{y}_c|\boldsymbol{\beta})\, d\mathbf{y}_c$$

$$= \log \int_{\mathbf{y}_c \in Y_c(\mathbf{y})} \frac{f_c(\mathbf{y}_c|\boldsymbol{\beta}^+).f(\mathbf{y}|\boldsymbol{\beta})}{f_c(\mathbf{y}_c|\boldsymbol{\beta})} \cdot \frac{f_c(\mathbf{y}_c|\boldsymbol{\beta})}{f(\mathbf{y}|\boldsymbol{\beta})}\, d\mathbf{y}_c$$

$$\geq \int \{\log(f_c(\mathbf{y}_c|\boldsymbol{\beta}^+).k(\mathbf{y}_c)) - \log(f_c(\mathbf{y}_c|\boldsymbol{\beta}).k(\mathbf{y}_c)) + \log f(\mathbf{y}|\boldsymbol{\beta}) \} \cdot \frac{f_c(\mathbf{y}_c|\boldsymbol{\beta})}{f(\mathbf{y}|\boldsymbol{\beta})} d\mathbf{y}_c \quad \text{using Jensen's inequality}$$

$$\geq \int \log f(\mathbf{y}|\boldsymbol{\beta}) \cdot \frac{f_c(\mathbf{y}_c|\boldsymbol{\beta})}{f(\mathbf{y}|\boldsymbol{\beta})} d\mathbf{y}_c$$

since $\boldsymbol{\beta}^+$ maximizes $\int \log(f_c(\mathbf{y}_c|\boldsymbol{\beta}^+).k(\mathbf{y}_c)).\frac{f_c(\mathbf{y}_c|\boldsymbol{\beta})}{f(\mathbf{y}|\boldsymbol{\beta})} d\mathbf{y}_c$

$$= \log f(\mathbf{y}|\boldsymbol{\beta})$$

with equality only when $f_c(\mathbf{y}_c|\boldsymbol{\beta}^+) = f_c(\mathbf{y}_c|\boldsymbol{\beta})$ almost everywhere. Therefore the EM algorithm steadily increases the likelihood and, provided the likelihood is bounded and there is only one local maximum of the likelihood, the algorithm must converge to it.

Even when the complete density is in the exponential family (6.1.4) the optimization in the M-step of (6.1.6) is of the same size as the original optimization. If this has no explicit solution, an iterative sub-algorithm must be used. This is avoided in another algorithm closely related to the EM algorithm. In the EM algorithm, $\boldsymbol{\beta}^+$ is chosen in each iteration to maximize the functions (6.1.2), (6.1.3), (6.1.5) or (6.1.6). However the convergence proof above is valid even if $\boldsymbol{\beta}^+$ is just chosen to increase the function rather than to maximize it, provided that $\boldsymbol{\beta}^+$ is chosen in such a way that it maximizes the function on convergence. If the maximization in each iteration is replaced by a single step of the NR, NRL or FS algorithms, the algorithm therefore retains the convergence properties of the EM algorithm. Dempster et al (1977) called this a GEM algorithm. Boyles (1983) further discusses convergence for GEM algorithms.

The guaranteed convergence of the EM and GEM algorithms is offset by the rate of convergence which is linear as opposed to the quadratic convergence of the NR algorithm (Dempster et al, 1977) and in some applications it has been found to be extremely slow. The main advantage of the EM and GEM algorithms lies in applications where they are substantially easier to implement then NR.

A disadvantage of the EM and GEM algorithms as opposed to NR, NRL or FS, is that they do not lead directly to estimates of the variances and covariances of the maximum likelihood estimator $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$. Louis (1982) however found an expression for $\partial^2 \log f(\mathbf{y}|\boldsymbol{\beta})/\partial\boldsymbol{\beta}^2$ based on the "complete" probability density $f_c(\cdot|\boldsymbol{\beta})$.

$$\frac{\partial \log f(\mathbf{y}|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = E\left[\frac{\partial \log f_c(\mathbf{y}_c|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \mid \mathbf{y}, \boldsymbol{\beta} \right]$$

$$\frac{\partial^2 \log f(\mathbf{y}|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2} = E\left[\frac{\partial^2 f_c(\mathbf{y}_c|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2} / f_c(\mathbf{y}_c|\boldsymbol{\beta}) \mid \mathbf{y}, \boldsymbol{\beta} \right] - \frac{\partial \log f(\mathbf{y}|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \cdot \frac{\partial \log f(\mathbf{y}|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}'}$$

$$= E\left[\frac{\partial^2 \log f_c(\mathbf{y}_c|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2} + \frac{\partial \log f_c(\mathbf{y}_c|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \cdot \frac{\partial \log f_c(\mathbf{y}_c|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}'} \mid \mathbf{y}, \boldsymbol{\beta}\right] - \frac{\partial \log f(\mathbf{y}|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \cdot \frac{\partial \log f(\mathbf{y}|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}'} \qquad (6.1.7)$$

At $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$, $\partial \log f(\mathbf{y}|\boldsymbol{\beta})/\partial\boldsymbol{\beta} = \mathbf{0}$ and so the second term above disappears

$$\frac{\partial^2 \log f(\mathbf{y}|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2} = E\left[\frac{\partial^2 \log f_c(\mathbf{y}_c|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2} + \frac{\partial \log f_c(\mathbf{y}_c|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \cdot \frac{\partial \log f_c(\mathbf{y}_c|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}'} \mid \mathbf{y}, \boldsymbol{\beta}\right] \qquad (6.1.8)$$

If Fisher's information matrix is required, the expectation must be made before $\hat{\boldsymbol{\beta}}$ is substituted for $\boldsymbol{\beta}$ and so the last term in (6.1.7) cannot be ignored; there is therefore no advantage over the standard formulae.

If $\mathbf{y}_c$ is in the exponential family (6.1.4),

$$\frac{\partial^2 \log f(\mathbf{y}|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2} = -\frac{\partial^2 \log a(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2} + E\left[\left(\mathbf{t}(\mathbf{y}_c) - \frac{\partial \log a(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right)\left(\mathbf{t}(\mathbf{y}_c) - \frac{\partial \log a(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right)' \mid \mathbf{y}, \boldsymbol{\beta} \right] \qquad (6.1.9)$$

The only additional quantity needed, in addition to those used in the EM algorithm, is $E[\mathbf{t}(\mathbf{y}_c)\mathbf{t}(\mathbf{y}_c)' \mid \mathbf{y}]$.

As with the EM algorithm itself, basing the evaluation of $\partial^2 \log f(\mathbf{y}|\boldsymbol{\beta})/\partial\boldsymbol{\beta}^2$ on the "complete" data is only better than direct evaluation if $f_c(\mathbf{y}_c|\boldsymbol{\beta})$ is "sufficiently simpler" in form than $f(\mathbf{y}|\boldsymbol{\beta})$.

In the following sections we examine several application areas of the EM algorithm that were identified by Dempster et al (1977) and, where possible, make comparisons with the algorithms developed in earlier chapters.

6.2 EXAMPLES : MISSING DATA

For the examples in this section, the "complete" data $\mathbf{y}_c$ is assumed to have independent components $y_1^c,\ldots,y_n^c$ which have log probability (density) functions of the form

$$\log f_i^c(y_i^c \mid \theta_i,\phi) = \alpha(\phi)\,(y_i^c\theta_i + g(\theta_i)) + h(\phi,y_i^c) \qquad (6.2.1)$$

where $\theta_i = \theta_i(\beta)$. This is the class of Generalized Linear Models (Nelder and Wedderburn, 1972) and their nonlinear extensions. It is assumed here that some of the observations are incompletely known because of grouping of some y_i^c so that it is only known that its value lies within a certain range, say $a_i < y_i < b_i$, or by complete loss of some such observations. The former includes probit models, in which an underlying normal response can be assumed (Finney, 1944) and models for censored survival data.

In this type of model, the "complete" and "incomplete" probability density functions both have two systematic parts θ_i and ϕ. The NR, NRL and FS algorithms developed in Chapters 3 and 4 can therefore be applied directly to the "incomplete" model. The iterations of the EM algorithm take the form,

E step : For each y_i^c whose value is incompletely known, find its expected value conditional on any information that is available about it and using the values of θ_i and ϕ from the previous iteration. If y_i^c is completely missing, then the expectation is $g'(\theta_i)$; otherwise, it is of the form $E[y_i^c \mid a_i < y_i^c < b_i]$.

M step : Maximize the complete-data log-likelihood with the incomplete observations replaced by their expected values from the E-step. Since ϕ can be factorized out, it can sometimes be ignored; however, it is often needed in the E-step.

If the "complete" model does not have an explicit maximum likelihood estimate then, rather than completing the maximization in the M-step, a GEM algorithm could be used, with the M-step being replaced by one iteration of NR, NRL or FS. These iterations of NRL and FS can always be performed as weighted least squares calculations; iterations of NR can also be evaluated as weighted least squares calculations if the parameters $\boldsymbol{\beta}$ are involved linearly in θ_i. In these cases, the GEM algorithm is also an IRLS algorithm.

There is only any point in including values y_i^c about which nothing is known if adding these extra observations makes the M-step easier. There are two main situations where this occurs. If extra unrecorded observations can be included in an unbalanced experimental design to make the "complete" design balanced, much simpler formulae can be used for the M-step than the least squares calculations which are necessary for the unbalanced data. Similarly, explicit parameter estimates can be easily found for many log-linear models for contingency tables which are complete cross-classifications of counts. If there are missing cells in a table, restoring the balance with unrecorded Poisson values avoids the lengthy calculations needed to apply NR to the recorded cells only.

The EM algorithm for grouped data is illustrated using probit models in which the observed response y_i is assumed initially to be a Bernoulli variable with probability of success $\pi_i = \Phi(\mathbf{x}_i'\boldsymbol{\beta})$ for $i=1,\ldots,n$ where $\Phi(\cdot)$ is the unit normal cumulative distribution function. These Bernoulli distributions can be generated from "complete" responses $y_1^c,\ldots,y_n^c$ which are independent normal variables $y_i^c \sim N(\mathbf{x}_i'\boldsymbol{\beta},1)$ if y_i is defined to be one when $y_i^c > 0$ and zero otherwise.

The EM algorithm therefore finds $\boldsymbol{\beta}^+$ in each iteration with a least squares calculation which has "explanatory" variables $\mathbf{x}_i$, weights $w_i = 1$ and "responses"

$$y_i^* = \begin{cases} E[\, y_i^c \mid y_i^c < 0 \,] & \text{if } y_i = 0 \\ E[\, y_i^c \mid y_i^c \geq 0 \,] & \text{if } y_i = 1 \end{cases}$$

$$= \begin{cases} \mathbf{x}_i'\boldsymbol{\beta} - \phi(\mathbf{x}_i'\boldsymbol{\beta}) \,/\, (1-\phi(\mathbf{x}_i'\boldsymbol{\beta})) & \text{if } y_i = 0 \\ \mathbf{x}_i'\boldsymbol{\beta} + \phi(\mathbf{x}_i'\boldsymbol{\beta}) \,/\, \phi(\mathbf{x}_i'\boldsymbol{\beta}) & \text{if } y_i = 1 \end{cases}$$

where $\phi(\cdot)$ is the unit normal probability density function.

If the observed response y_i is a binomial count of successes in r_i trials where the probability of success is given by the same probit function, the same least squares calculation could be done with r_i observations generated corresponding to each y_i using the above method. These can however be combined into a single observation with "explanatory" variables $\mathbf{x}_i$, weight $w_i = r_i$ and "response"

$$y_i^* = \mathbf{x}_i'\boldsymbol{\beta} + \phi(\mathbf{x}_i'\boldsymbol{\beta}) \left\{ \frac{y_i}{r_i\phi(\mathbf{x}_i'\boldsymbol{\beta})} - \frac{r_i - y_i}{r_i(1-\phi(\mathbf{x}_i'\boldsymbol{\beta}))} \right\} .$$

The EM algorithm is therefore of similar complexity to the NR and FS algorithms described in Section 3.5. Starting values can be found in a similar way to the method described for the algorithms in Section 3.5 since the iterations only depend on the previous value of $\boldsymbol{\beta}$ through the systematic part $\mathbf{x}_i'\boldsymbol{\beta}$. An iteration of the algorithm with $\mathbf{x}_i'\boldsymbol{\beta}$ replaced by $\Phi^{-1}(y_i/r_i)$ can be used, with minor adjustments when $y_i = 0$ or $y_i = r_i$.

The EM algorithm was applied to the artificial data in Table 3.5.1, and its iterations are shown in Table 6.2.1. The EM algorithm performs considerably worse than the NR and FS algorithm which were applied to the same data in Section 3.5 and whose iterations are given in Table 3.5.2.

In most situations where data is grouped, iterations of the EM and GEM algorithms are not significantly simpler than iterations of the NR or NRL algorithm. In view of the superior convergence of the NR and usually also NRL algorithms these would usually be preferred to the EM algorithm for this type of problem.

TABLE 6.2.1

Iterations of EM Algorithm Applied to Data in Table 3.5.1.
Starting Values (Iteration 0) were Found as Decribed in the Text.

Iteration	β_1	β_2	Log-Likelihood	Convergence Rate, C
0	-0.51193	0.42002	-1.60771	
1	-0.52389	0.42899	-1.60082	0.59359
2	-0.53115	0.43411	-1.59847	0.59519
3	-0.53554	0.43707	-1.59766	0.59594
4	-0.53819	0.43880	-1.59738	0.59624
5	-0.53978	0.43981	-1.59728	0.59633
6	-0.54074	0.44041	-1.59724	0.59631
7	-0.54132	0.44077	-1.59723	0.59624
8	-0.54166	0.44097	-1.59722	0.59615
9	-0.54187	0.44110	-1.59722	0.59606
10	-0.54199	0.44117	-1.59722	0.59596
11	-0.54206	0.44122	-1.59722	0.59586
12	-0.54211	0.44124	-1.59722	0.59572

6.3 EXAMPLES : MIXTURES AND TOTALS

In this section we first consider examples where the i'th of n independent responses, y_i, has a distribution with probability (density) function of the form

$$f_i(y_i \mid \boldsymbol{\beta},\boldsymbol{\pi}) = \sum_{j=1}^{q} \pi_j f_{ij}(y_i \mid \boldsymbol{\beta})$$

and where $f_{ij}(\cdot|\boldsymbol{\beta})$ are probability (density) functions and $\sum\pi_j = 1$. This type of model usually arises when it is uncertain which of q distributions the response y_i came from. An important application of these mixture models is in discrimination problems where there are some responses that are classified into one of q populations and others that are unclassified, with unknown proportions $\{\pi_j\}$ from the q populations.

This type of model usually cannot be written with a small number of systematic parts and so the NRL algorithms in Chapter 4 are usually impractical. Dempster et al (1977) suggested defining the complete data for y_i to be $(y_i,\mathbf{z}_i)$ where $\mathbf{z}_i$ is a vector of q 0/1 indicator variables which denote the component of the mixture from which y_i was sampled. Then $(y_i,\mathbf{z}_i)$ has a joint distribution with log joint probability function that can be written as

$$\log f_i^c(y_i,\mathbf{z}_i \mid \boldsymbol{\beta},\boldsymbol{\pi}) = \sum_j \{ z_{ij} \log f_{ij}(y_i \mid \boldsymbol{\beta}) + z_{ij} \log \pi_j \} \quad (6.3.1)$$

The E step of the EM (or GEM) algorithm therefore evaluates

$$z_{ij}^* = E[z_{ij} \mid y_i,\boldsymbol{\beta}] = f_{ij}(y_i|\boldsymbol{\beta})/(\sum_j f_{ij}(y_i|\boldsymbol{\beta})$$

and the M step maximizes (or takes a NR, NRL or FS step to increase) the "complete" log-likelihood (6.3.1) with $\{z_{ij}\}$ replaced by $\{z_{ij}^*\}$. The "complete" log-likelihood is the sum of two parts, the first of which depends only on $\boldsymbol{\beta}$ and the second only on $\boldsymbol{\pi}$, so that both can be separately maximized. The first part can often be written with one or two systematic parts which allows the M-step to be performed with least

squares calculations using the methods of Chapters 3 and 4. Maximizing the second part leads to $\pi_j^+ = \sum_i z_{ij}^*/n$.

A related kind of mixing occurs when two or more Poisson counts are combined (such as when 2 or more cells in a contingency table are collapsed together). The count y_i is then Poisson with parameter $\lambda_i = \sum_j \lambda_{ij}$ which is the sum of the rates of the component counts. If many cells are collapsed together and a log-linear model for the component counts is used, the model cannot be written with a small number of systematic parts. The "complete" data can be defined by replacing y_i with a vector $\mathbf{z}_i$ which holds the component Poisson counts. The E-step of the EM or GEM algorithm evaluates $E[z_{ij}|y_i] = y_i\lambda_{ij}/\sum_j \lambda_{ij}$ and the M-step is performed as though the component counts $\{z_{ij}\}$ were known. If the complete data is a balanced contingence table, explicit solutions may be available. Otherwise an iteration of NR, NRL or FS can usually be implemented with the IRLS algorithms of Chapters 3 and 4. A similar method can be used whenever an observed y_i is the total of unobserved values, $y_i = \sum_j z_{ij}$, and where $\{z_{ij}\}$ satisfy a model of the form (6.2.1).

6.4 EXAMPLES : VARIANCE COMPONENTS

In this section we consider normal mixed linear models with both random and fixed effects. Dempster et al (1977) showed that the EM algorithm could be easily applied to this type of model if the random effects and the observed data are together denoted the "complete" data. The method is illustrated below using models with a single random effect, which can be written in the form (4.4.1). We define $\mathbf{y}_c' = [\mathbf{y}' : \mathbf{b}']$. Then the elements of $\mathbf{b}$ given $\mathbf{y}$ are independent with

$$b_i \mid \mathbf{y} \sim N\left(\frac{\phi_b}{r_i\phi_b + \phi_e} (Y_i - \mathbf{X}_i'\boldsymbol{\beta}) , \frac{\phi_b \phi_e}{\phi_e + r_i\phi_b} \right)$$

where Y_i and $\mathbf{X}_i$ are the totals of the responses and explanatory variables from the observations in block i. Since

$$f_c(\mathbf{y}_c|\boldsymbol{\beta},\phi_e,\phi_b) = -\{ n.\log \phi_e + a.\log \phi_b + \sum\sum(y_{ij}-b_i-\mathbf{x}_{ij}'\boldsymbol{\beta})^2/\phi_e + \sum b_i^2 / \phi_b \}/2 + \text{const}$$

(6.1.2) is equivalent to maximizing with respect to $\boldsymbol{\beta}^+$, ϕ_e^+ and ϕ_b^+ the function

$$-n.\log \phi_e - \phi_e^{+-1} \{ \sum\sum (y_{ij} - \frac{\phi_b(Y_i-\mathbf{X}_i'\boldsymbol{\beta})}{r_i\phi_b+\phi_e} - \mathbf{x}_{ij}'\boldsymbol{\beta}^+)^2 + \sum \frac{r_i\phi_e\phi_b}{r_i\phi_b+\phi_e} \}$$

$$-a.\log \phi_b^+ - \phi_b^{+-1} \{ \sum \frac{\phi_b^2(Y_i-\mathbf{X}_i'\boldsymbol{\beta})^2}{(r_i\phi_b+\phi_e)^2} + \sum \frac{\phi_b\phi_e}{r_i\phi_b+\phi_e} \}$$

$\boldsymbol{\beta}^+$ can be found with an unweighted least squares calculation and ϕ_e^+ and ϕ_b^+ can be found by dividing the two terms in braces by n and a respectively. The EM calculations can be considerably speeded up if $\mathbf{y}$ is initially transformed to $\mathbf{y}^0$ as described in Section 4.4 since each block effect b_i is only involved in one element of $\mathbf{y}^0$ in the model for the transformed data.

Starting values for this algorithm are harder to obtain than for the NRL algorithms in Section 4.4. The most "obvious" starting value

of $\phi_b = 0$ with ϕ_e and β set from unweighted least squares cannot be used since the EM iterations can never move ϕ_b away from zero. (The convergence proof in Section 6.1 fails because $\phi_b = 0$, which is the maximum in the M-step, is not at a turning point of the function being maximized). A fixed effects analysis of variance can however be performed and ϕ_b can be set equal to the mean sum of squares of the fixed block effects to get starting values. This can however be expensive if there are many blocks in the model. Iterations of the EM algorithm are comparable in complexity to those of the FS-alternation algorithm described in Section 4.4.

TABLE 6.4.1

Iterations of the EM Algorithm Applied to Data in Table 4.4.2 Starting Values (Iteration 0) were Those Used as Starting Values for the FS-alternation Algorithm in Table 4.4.3.

Iteration	μ_1	μ_2	ϕ_e	ϕ_b	Log-Likelihood	Convergence Rate, C
0	2.84843	4.87078	0.00127	1.78929	-13887.333	
1	2.84843	4.87078	1.96307	2.88107	-19.80374	0.28348
2	2.90312	4.83489	2.28052	2.67140	-19.66178	0.39032
3	2.92808	4.82591	2.33549	2.57163	-19.65539	0.51890
4	2.94040	4.82700	2.34698	2.53237	-19.65458	0.65151
5	2.94804	4.83085	2.34996	2.51759	-19.65429	0.75625
6	2.95366	4.83513	2.35090	2.51180	-19.65411	0.81654
7	2.95820	4.83917	2.35127	2.50925	-19.65398	0.84479
8	2.96203	4.84277	2.35145	2.50789	-19.65388	0.85679
9	2.96531	4.84593	2.35155	2.50703	-19.65381	0.86173
10	2.96814	4.84868	2.35163	2.50639	-19.65375	0.86379
11	2.97060	4.85106	2.35169	2.50590	-19.65371	0.86469
⋮	⋮	⋮	⋮	⋮	⋮	⋮
23	2.98364	4.86382	2.35200	2.50362	-19.65359	0.86614
24	2.98402	4.86420	2.35201	2.50356	-19.65359	0.86617
25	2.98435	4.86452	2.35202	2.50351	-19.65359	0.86620
26	2.98463	4.86480	2.35203	2.50347	-19.65359	0.86622

To illustrate the algorithm, it was applied to the data in Table 4.4.1. Since the intention was to compare the EM algorithm with the algorithms presented in Section 4.4, the starting values found for the

FS-alternation algorithm in Table 4.4.3 were again used. The iterations are shown in Table 6.4.1. The EM algorithm performed considerably worse than any of the algorithms examined in Section 4.4.

The EM algorithm can be applied in a similar way to any heteroscedastic model where $y_1,\ldots,y_n$ are independent with $y_i \sim N(\mathbf{x}_i^{(1)}\boldsymbol{\beta}^{(1)}, \mathbf{x}_i^{(2)}\boldsymbol{\beta}^{(2)})$. The "complete" data can then be defined to be $\mathbf{y}_c' = [\mathbf{y}' : \mathbf{B}]$ where $\mathbf{y} = \mathbf{x}_i^{(1)}\boldsymbol{\beta}^{(1)} + \sum_j \sqrt{x_{ij}^{(2)}} B_{ij}$, all elements of the matrix $\mathbf{B}$ are independent and $B_{ij} \sim N(0, \boldsymbol{\beta}_j^{(2)})$. The EM algorithm then has a similar form to the EM algorithm above for variance components.

The EM algorithm is not recommended for the applications above due to the difficulty in finding starting values, poor convergence and the problems in finding estimates of the variances of the maximum likelihood estimates. However it can be applied to other problems for which NRL algorithms cannot be so easily used. For example, the EM algorithm can be applied to models with two or more block effects and a variation can be used to find REML estimators (Dempster et al, 1977 and Dempster et al, 1984). However see Harville (1977) for implementation details of FS algorithms in these types of model; they would be expected to have better convergence properties than the EM algorithm.

6.5 EXAMPLES : HYPERPARAMETER ESTIMATION

The distribution of some responses y_i can be generated as a compound distribution, with a known conditional distribution for y_i given a "hyperparameter" p_i and a known marginal distribution for the "hyperparameter" p_i which depends on the unknown parameters $\boldsymbol{\beta}$. Common examples that can be expressed in this form are the negative binomial and beta-binomial distributions, which can be expressed using the notation of Johnson and Kotz (1969) as

$$\text{Poisson}(m_i) \overset{m_i}{\wedge} \text{Gamma}(\phi_i\ ,\ \mu_i/\phi_i) \tag{6.5.1}$$

and

$$\text{binomial}(r_i,\ p_i) \overset{p_i}{\wedge} \text{beta}(\alpha_i,\ \beta_i) \tag{6.5.2}$$

respectively. Dempster et al (1977) suggested applying the EM algorithm to this type of problem by including the "hpyerparameters" $\mathbf{p}$ with $\mathbf{y}$ to give the "complete" data.

We shall use the negative binomial distribution (6.5.1) to illustrate the method. The "complete" distribution is then

$$f_c(\mathbf{y}_c|\boldsymbol{\phi},\boldsymbol{\mu}) = \prod_{i=1}^{n} m_i^{y_i}\ \frac{\exp(-m_i)}{y_i}\ .\ (m_i\phi_i/\mu_i)^{\phi_i}.\ \frac{\exp(-m_i\phi_i/\mu_i)}{\Gamma(\phi_i)\ m_i}\ .$$

The conditional distributions of $m_1,\ldots,m_n$ given $\mathbf{y}$ are independent with

$$m_i \mid \mathbf{y}\ \sim\ \text{Gamma}\ (\ y_i+\phi_i\ ,\ \frac{\mu_i}{\mu_i+\phi_i}\ )$$

and (6.1.2) is equivalent to maximizing with respect to $\boldsymbol{\mu}^+$ and $\boldsymbol{\phi}^+$ the function

$$\sum_{i=1}^{n} \{\ \phi_i^+\ E[\log m_i|y_i,\boldsymbol{\phi},\boldsymbol{\mu}] - \phi_i^+/\mu_i^+\ E[m_i|y_i,\boldsymbol{\phi},\boldsymbol{\mu}] - \log\Gamma(\phi_i^+) - \phi_i^+\log(\mu_i^+/\phi_i^+)\ \} \tag{6.5.3}$$

where $E[\log m_i \mid y_i, \boldsymbol{\phi}, \boldsymbol{\mu}] = \psi'(y_i + \phi_i) + \log\left(\frac{\mu_i}{\mu_i + \phi_i}\right)$,

$$E[m_i \mid y_i, \boldsymbol{\phi}, \boldsymbol{\mu}] = \mu_i \cdot \frac{(y_i + \phi_i)}{(\mu_i + \phi_i)}$$

and $\psi'(x)$ is the digamma function $\partial \log \Gamma(x)/\partial x$. However for any parameterization, such as $\mu_i = \exp(\mathbf{x}_i'\boldsymbol{\beta})$ and $\phi_i = \phi$, the maximization at each step in (6.5.3) is not significantly simpler than the whole original negative binomial optimization. A NR or FS algorithm would have to be used at each EM step and this would also involve digamma and trigamma functions. Clearly the EM algorithm offers no advantage over direct use of NR, NRL or FS in negative binomial models.

A similar result holds when the beta-binomial distribution is expressed as the compound distribution (6.5.2). Each iteration of the EM algorithm involves an optimization for which NR, NRL or FS must be used and each of these sub-iterations requires evaluation of digamma and trigamma functions. Again there is no advantage in using the EM algorithm over direct application of the NR, NRL or FS algorithms to the beta-binomial log-likelihood.

Though the EM algorithm may have benefits in some problems involving hyperparameters, in models where the methods of Chapters 3 and 4 can be applied, the latter are usually to be preferred.

6.6 EXAMPLES : ROBUST ESTIMATION IN LINEAR MODELS

In this section we start by reconsidering maximum likelihood estimation in models where the independent responses $y_1,\ldots,y_n$ have log probability density functions

$$\log f_i(y_i \mid \eta_i,\sigma^2) = -\phi\left(\frac{y_i-\eta_i}{\sigma}\right) - \log\sigma - \log\{\int_{-\infty}^{\infty}\exp(-\psi(z))dz\}$$

where $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$, $\psi(\cdot)$ is a symmetric function and the distribution is longer-tailed than the normal distribution.

Subject to minor regularity conditions any such symmetric probability density function can be expressed in the form

$$f_i(y_i|\boldsymbol{\beta},\sigma^2) = \int_0^{\infty}\sqrt{\left\{\frac{q_i}{2\pi\sigma^2}\right\}}\exp\left(-q_i\frac{(y_i-\mathbf{x}_i'\boldsymbol{\beta})^2}{2\sigma^2}\right) m(q_i)\,dq_i \qquad (6.6.1)$$

for some $m(\cdot)$. Note that the function $f_i(\cdot)$ is closely related to a Laplace transform of the function $m(\cdot)$. By integrating both sides of (6.6.1) with respect to y_i it can be easily seen that $\int_0^{\infty} m(q_i)\,dq_i = 1$. If $m(q_i)\geq 0$ for all q_i then the distribution of y_i can be generated as the ratio of a $N(\mathbf{x}_i'\boldsymbol{\beta},\sigma^2)$ variable to the square root of an independent variable with probability density function $m(q_i)$. These latter distributions were called N/I distributions by Dempster et al(1980); Andrews and Mallows(1974) showed that they covered many commonly used long-tailed alternatives to the normal distribution such as the Student's t, Cauchy, double exponential and logistic distributions.

We can however apply the EM algorithm to $f_i(y_i|\boldsymbol{\beta},\sigma^2)$ using (6.6.1) whether or not $m(\cdot)$ is non-negative since it can be written in form (6.1.1) with $\mathbf{y}_c' = [\mathbf{y}':\mathbf{q}']$,

$$f_c(\mathbf{y}_c \mid \boldsymbol{\beta},\sigma) = (2\pi\sigma^2)^{-n/2}\prod_{i=1}^{n} q_i^{-1/2}\exp\left(-q_i\frac{(y_i-\mathbf{x}_i'\boldsymbol{\beta})^2}{2\sigma^2}\right) m(q_i)$$

and $k(\mathbf{y}_c) = \prod m(q_i)^{-1}$. Then (6.1.2) is equivalent to maximizing with respect to $\boldsymbol{\beta}^+$ and σ^+ the function

$$-n.\log\sigma^{+} - \frac{1}{\sigma^{+2}} \sum_{i=1}^{n} w_i(\beta,\sigma).(y_i - x_i'\beta^{+})^2 \qquad (6.6.2)$$

where $w_i(\beta,\sigma) = \int_0^\infty \frac{q_i^{3/2}}{\sqrt{(2\pi)}\sigma} \exp\left(\frac{-q_i(y_i-x_i'\beta)^2}{2\sigma^2}\right) m(q_i)\, dq_i \;/\; f_i(y_i|\beta,\sigma^2)$

$$= -\frac{\sigma^2 f_i'(y_i \mid \beta,\sigma^2)}{(y_i-x_i'\beta)\, f_i(y_i \mid \beta,\sigma^2)}$$

$$= \sigma \frac{\psi'\left((y_i-x_i'\beta)/\sigma\right)}{(y_i-x_i'\beta)}$$

It should be noted that the algorithm does not need $m(\cdot)$ to be explicitly found. Once the weights $w_i(\beta,\sigma)$ have been evaluated, the optimization of (6.6.2) is a simple explicit least squares calculation and the EM algorithm is therefore an IRLS algorithm.

The formulae are illustrated with the log-tailed distribution (4.5.2). For the EM algorithm the weights are

$$w_i = \begin{cases} 1 & |y_i-x_i'\beta|/\sigma < \phi \\ \dfrac{\phi^2\sigma^2}{(y_i-x_i'\beta)^2} & |y_i-x_i'\beta|/\sigma \geq \phi \end{cases}$$

The NR-alternation and FS-alternation IRLS formulae given in Section 4.6 are similar in complexity to the EM formulae but the NR-alternation algorithm involves negative weights.

The EM algorithm was applied to the data in Table 4.6.1 and its iterations are shown in Table 6.6.1. The EM algorithm converges extremely slowly. The NRL-alternation and NR algorithms, whose iterations are given in Table 4.6.2, perform much better.

If NR, NRL or FS algorithms are used, the variance-covariance matrix of $\hat{\beta}$ can be estimated by $(X'WX)^{-1}$ from the last IRLS iteration for β. Log $\hat{\sigma}$ is asymptotically independent and its variance can also be estimated from quantities used in the last iteration for these algorithms, as described in Section 4.6. The EM variance formulae (6.1.8) which were derived by Louis(1982) do not provide a simpler

TABLE 6.6.1

Iterations of EM Algorithm Applied to Fit Log-tail Model with $\phi = 1.3$ to Data in Table 4.6.1. Starting Values (Iteration 0) were Found from Ordinary Least Squares.

Iteration	β_0	β_1	β_2	β_3	log σ	Log-Likelihood	Convergence Rate, C
0	-39.91967	0.71564	1.29529	-0.15212	1.07096	-31.41310	
1	-41.16943	0.84744	0.86200	-0.12307	0.79857	-28.23576	1.08410
2	-41.12496	0.88827	0.69196	-0.11127	0.61061	-26.87757	0.92220
3	-39.77449	0.87892	0.61888	-0.10387	0.46930	-25.74951	0.65283
4	-38.31231	0.86008	0.57220	-0.09763	0.32625	-24.64932	0.43612
5	-37.34304	0.84767	0.54131	-0.09358	0.20045	-23.93669	0.72827
6	-36.98684	0.84331	0.52058	-0.09018	0.10453	-23.56842	1.25461
7	-36.89116	0.84293	0.50822	-0.08839	0.03598	-23.38216	1.01128
8	-36.99481	0.84497	0.49722	-0.08606	-0.01955	-23.26077	0.85435
9	-37.14177	0.84697	0.48859	-0.08361	-0.06616	-23.17464	0.79817
10	-37.28021	0.84850	0.48174	-0.08131	-0.10495	-23.11404	0.77009
11	-37.39652	0.84960	0.47638	-0.07934	-0.13677	-23.07297	0.75124
12	-37.48781	0.85040	0.47230	-0.07777	-0.16244	-23.04632	0.73900
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
33	-37.73245	0.85287	0.46103	-0.07373	-0.24884	-23.00570	0.74186
34	-37.73254	0.85288	0.46103	-0.07373	-0.24889	-23.00570	0.74189
35	-37.73261	0.85288	0.46102	-0.07372	-0.24892	-23.00570	0.74191
36	-37.73266	0.85288	0.46102	-0.07372	-0.24895	-23.00570	0.74193
37	-37.73270	0.85288	0.46102	-0.07372	-0.24897	-23.00570	0.74194

method of calculating the second derivatives of the log-likelihood in this example so that if variances are required, computations similar to those of a single NR, NRL or FS iteration must be done even for the EM algorithm. There seems therefore little to commend the EM algorithm for this type of model.

Usually the models used for robust fitting of linear models have a further parameter ϕ which can be adjusted to give varying amounts of robustness. We next consider adaptive models of the form

$$\log f_i(y_i|\boldsymbol{\beta},\sigma^2,\phi) = -\psi\left(\frac{y_i-\mathbf{x}_i'\boldsymbol{\beta}}{\sigma},\phi\right) - \log\sigma - \log\left(\int_{-\infty}^{\infty}\exp\{-\psi(z,\phi)\}dz\right) \tag{6.6.3}$$

where β, σ and ϕ are all treated as unknown parameters. As before it is possible to write

$$f_i(y_i|\beta,\sigma^2,\phi) = \int_0^\infty (2\pi\sigma^2)^{-1/2} q_i^{1/2} \exp\left(\frac{-q_i(y_i-x_i'\beta)^2}{2\sigma^2}\right) m(q_i,\phi)\, dq_i$$

for some $m(q_i,\phi)$. For a general symmetric distribution, it is not always possible to find $k(q_i)$ such that $m(q_i,\phi).k(q_i) \geq 0$ for all q_i and ϕ. We therefore restrict attention to N/I distributions (Andrews and Mallows, 1974) where $m(q_i,\phi) \geq 0$. A step of the EM algorithm can be expressed as

(1) maximize with respect to β^+ and σ^+ the function

$$-n \log \sigma^+ - \sum w_i(\beta,\sigma,\phi).(y_i-x_i'\beta^+)^2 / (2\sigma^{+2})$$

(2) maximize with respect to ϕ^+ the function

$$-\sum \frac{\int_0^\infty \log m(q_i,\phi^+)\, q_i^{1/2} \exp\{\frac{-q_i(y_i-x_i'\beta)^2}{2\sigma^2}\}\, m(q_i,\phi)\, dq_i}{f_i(y_i|\beta,\sigma^2,\phi)}$$

where

$$w_i(\beta,\sigma,\phi) = \frac{\sigma\, \psi'((y_i-x_i'\beta)/\sigma\ ,\ \phi)}{(y_i-x_i'\beta)} \quad .$$

Though each EM iteration for β and σ^2 is an ordinary weighted least squares calculation, the iteration for ϕ is now considerably more complex and cannot be done unless the function $m(\cdot,\cdot)$ is known. In distributions such as the log-tailed distribution used above, the EM algorithm cannot therefore be used for adaptive estimation.

We therefore concentrate here on one particular example where $m(\cdot,\cdot)$ is known, which is where $(y_i-x_i'\beta)/\sigma$ has a t-distribution with ϕ degrees of freedom so that $m(\cdot,\phi)$ is a gamma probability density function. It can then be shown that

$$w_i(\beta,\sigma,\phi) = \frac{\phi+1}{\phi + (y_i-x_i'\beta)^2/\sigma^2}$$

and step (2) for ϕ can be expressed as

(2) maximize with respect to ϕ^+ the function

$$\frac{\phi^+-1}{2}\sum u_i(\beta,\sigma,\phi)-\frac{\phi^+}{2}\sum w_i(\beta,\sigma,\phi)-n\{\psi(\frac{\phi^+}{2})+\frac{\phi^++1}{2}\log 2-\frac{\phi^+}{2}\log\phi\}$$

where

$$u_i(\beta,\sigma,\phi) = \psi'(\frac{\phi+1}{2}) - \log(\frac{\phi + (y_i-x_i'\beta)^2/\sigma^2}{2})$$

and $\psi(\cdot)$ is the log gamma function. This clearly cannot be explicitly solved so that a separate iterative univariate optimization is needed. NR applied to this sub-problem leads to sub-iterations of the form

$$\phi^+ = \phi + \frac{\bar{U} - \bar{W} - \log 2 - \psi'(\phi/2) + 1 + \log\phi}{\psi''(\phi/2)/2 - \phi^{-1}}$$

where $\bar{U}$ and $\bar{W}$ are the averages of $u_i(\beta,\sigma,\phi)$ and $w_i(\beta,\sigma,\phi)$. The complexity of the EM iteration for ϕ clearly detracts from the simplicity of the iteration for β and σ^2. A GEM algorithm would only take a single NR sub-iteration for ϕ here.

Iterations of the EM and GEM algorithms, applied to the artificial data in Table 4.6.4, are shown in Table 6.6.2. The NRL-alternation algorithm, whose iterations are given in Table 4.6.5, converges considerably faster and has the advantage that it can easily provide estimates of the variances and covariances of the maximum likelihood estimators.

TABLE 6.6.2

Iterations of EM and GEM Algorithms Applied to Fit a T-distribution to Data in Table 4.6.1. Starting Values (Iteration 0) were $\phi = 5.0$ and the Ordinary Least Squares Estimates of μ and σ^2

Iteration	Mean, μ	log σ	ϕ	Log-Likelihood	Convergence Rate, C
EM Algorithm					
0	-0.46760	0.76450	5.00000	-16.37611	
1	-0.15111	0.62690	5.41109	-15.95800	0.99239
2	-0.05175	0.56074	5.39618	-15.87200	0.96784
3	-0.00419	0.52404	5.17365	-15.83039	0.94350
4	0.02973	0.49677	4.86649	-15.78842	0.92810
5	0.06095	0.47073	4.52719	-15.73822	0.91660
6	0.09296	0.44248	4.17748	-15.67692	0.90593
7	0.12677	0.41039	3.82794	-15.60228	0.89478
8	0.16236	0.37352	3.48581	-15.51251	0.88264
9	0.19907	0.33138	3.15766	-15.40693	0.86947
10	0.23578	0.28395	2.85008	-15.28701	0.85551
11	0.27103	0.23187	2.56936	-15.15725	0.84131
12	0.30332	0.17656	2.32066	-15.02524	0.82766
⋮	⋮	⋮	⋮	⋮	⋮
48	0.43251	-0.26334	1.27521	-14.48209	0.80113
49	0.43252	-0.26338	1.27516	-14.48209	0.80114
50	0.43252	-0.26342	1.27512	-14.48209	0.80114
51	0.43253	-0.26344	1.27508	-14.48209	0.80115
52	0.43253	-0.26346	1.27505	-14.48209	0.80115
GEM Algorithm					
0	-0.46760	0.76450	5.00000	-16.37611	
1	-0.15111	0.62690	5.37829	-15.95791	0.98659
2	-0.05047	0.55994	5.36644	-15.86990	0.96782
3	-0.00241	0.52276	5.13513	-15.82683	0.94136
4	0.03223	0.49483	4.80958	-15.78211	0.92362
5	0.06473	0.46759	4.44819	-15.72751	0.91027
6	0.09858	0.43746	4.07558	-15.65982	0.89781
7	0.13465	0.40271	3.70410	-15.57643	0.88468
8	0.17271	0.36233	3.34280	-15.47548	0.87041
9	0.21178	0.31590	3.00044	-15.35697	0.85513
10	0.25034	0.26364	2.68578	-15.22420	0.83946
11	0.28652	0.20673	2.40654	-15.08460	0.82440
12	0.31857	0.14734	2.16790	-14.94875	0.81115
⋮	⋮	⋮	⋮	⋮	⋮
47	0.43251	-0.26333	1.27522	-14.48209	0.80113
48	0.43252	-0.26338	1.27517	-14.48209	0.80114
49	0.43252	-0.26341	1.27512	-14.48209	0.80114
50	0.43253	-0.26344	1.27509	-14.48209	0.80115
51	0.43253	-0.26346	1.27506	-14.48209	0.80115

6.7 CONCLUDING REMARKS ABOUT THE EM ALGORITHM

The EM algorithm can be applied to many different types of problem, but it has the disadvantage that its convergence can sometimes be extremely slow. This in turn can lead to the algorithm being prematurely stopped before the maximum likelihood estimate has been reached. With a single unknown parameter, Aitken's acceleration method might be used to improve convergence. However the NR algorithm (and usually the NRL and FS algorithms) converges much faster near the maximum likelihood estimate and, where it can be easily implemented, it would usually be preferred to the EM (or GEM) algorithm.

Another disadvantage of the EM algorithm is that it does not easily lead to consistent estimates of the variances of the maximum likelihood estimators, whereas these can be easily found when NR, NRL or FS are used.

However in Sections 6.2 to 6.6, several types of models have been identified where the calculations for an EM iteration are considerably easier to implement than those for NR (or NRL or FS). In these examples, the EM algorithm provides a convenient estimation method.

7. CONCLUSION

In models with one, two or three systematic parts, there are often several algorithms for maximum likelihood parameter estimation that can be implemented with sequences of least squares computations; these are called iteratively reweighted least squares (IRLS) algorithms. The class of NRL algorithms described in Chapters 3 and 4 contains algorithms of this form and often includes the Newton-Raphson (NR) algorithm and Fisher's scoring technique (FS). Different algorithms in the NRL class often have different properties and the requirements of fast asymptotic convergence and good initial iterations are not always shared by the same algorithm. NRL algorithms that behave like FS are often best in the initial iterations, whereas ones that behave like NR are best near the maximum likelihood estimate.

For some models, the FS algorithm is not in the NRL class, but can still be implemented as an IRLS algorithm. The EM algorithm which was described in Section 6.1, can be used for some models and is rarely in the NRL class; it can sometimes also be implemented as an IRLS algorithm. The FS and EM algorithms often have slow asymptotic convergence rates and the EM algorithm in particular can sometimes be extremely slow to converge.

A thorough investigation of the relative performances of the various IRLS algorithms has not been undertaken in this thesis. For each particular type of model examined in the thesis, only a few illustrative data sets have been analysed, but an extensive set of examples would be needed before any firm recommendations could be made.

There are two advantages in using algorithms that can be implemented as a sequence of least squares calculations. Firstly, QR

algorithms can be used for the least squares calculations and this allows the maximum likelihood estimate to be more accurately determined in examples where the maximum likelihood estimators are highly correlated. Secondly, a wide range of models can be estimated by algorithms which fit into the same framework, allowing them all to be implemented within one computer package. For example, all 31 IRLS algorithms that have been used to produce the various tables in this thesis have all been implemented with a single Fortran program in which only a few subroutines had to be changed for different models.

To show the amount of work needed from the user of such a general system to implement a new IRLS algorithm for a different model, the necessary subroutines in the author's program are described next. The subroutines all have access to the following variables in a common block.

N	number of log-likelihood components
Y(N)	array holding responses
X(N,NX)	array holding other information about the log-likelihood components, such as the explanatory variables or the number of binomial trials
BETA1(.), BETA2(.)	arrays of size P(1) and P(2) holding two subsets of the parameters
P(2)	array giving sizes of two subsets of parameters. P(2) = 0 unless an alternation algorithm is used
LP1(N), LP2(N)	arrays that the user may use to hold systematic parts
IALG	integer indicating which algorithm to use if several are possible
PART	in alternation algorithms, PART indicates which part of the iteration is being performed (1 or 2)

N, Y, NX, X and IALG are set up interactively by commands read by the core of the program since they are not specific to any algorithm or model. All algorithms require the following four subroutines to be written by the user:--

SUBROUTINE GETXWZ (ROW, NPARAM, W, Z, I, PSEUDO)

Sets up array ROW(NPARAM), W, and Z as the explanatory variables, weights and response for the I'th log-likelihood component. The weight and response are usually functions of the I'th elements in the arrays LP1 and LP2. The explanatory variables are usually set up from the I'th row of array X. If pseudo-observations are needed for the I'th log-likelihood component, parameter PSEUDO, which is normally zero, is set to one or two and GETXWZ will be called again with the same I.

REAL FUNCTION ALLIKE (PARAMS, IP)

Returns the log-likelihood corresponding to parameters in the arrays PARAMS and BETA2 if PART=1 or corresponding to BETA1 and PARAMS if PART=2. It also sets up values of LP1 and LP2 if these will be used by GETXWZ. If an adjustment algorithm is used, ALLIKE initially adjusts PARAMS.

SUBROUTINE INITP

Sets up starting values for parameters and systematic parts either by initializing all parameters and calling ALLIKE or by initializing some parameters and the systematic parts.

SUBROUTINE AUGMENT (R, IR, NPARAM, DIMAX, ODMAX, IFAULT)

If subroutine NUISNC is to be used to include an extra nuisance parameter, this is done by AUGMENT. In that case, GETXWZ should set Z=0.0 and the last element of ROW to the response. Otherwise, AUGMENT should return without effect.

If alternation algorithms are used, two extra subroutines GETXWZ2 and AUGMENT2 must be provided to perform the same functions as GETXWZ and AUGMENT for the second set of parameters BETA2.

The intention here is not to recommend the actual program that has been used, but to indicate the amount of programming required for each type of algorithm in a general IRLS system. Clearly more would be required of the user to implement an algorithm for a new type of model than in packages such as GLIM where a few macros in the command

language are all that is needed, but a much wider class of models and algorithms could be implemented within such a system.

Finally, it is noted that the IRLS algorithms developed in the thesis do not treat fully the problem of inequality constraints. Constraints of this kind are very common. For example Poisson means, normal variances, degrees of freedom for t-distributions and negative binomial auxiliary parameters must all be non-negative. In many models, these constraints are relatively unimportant since the maximum likelihood estimate will not satisfy the constraints as equalities. However, in other models, it is possible for the maximum likelihood estimate to be on a boundary of the set of feasible values of the parameters. Development of modified IRLS algorithms for use with this type of problem is an area for further research.

APPENDIX A : FORTRAN SUBROUTINES FOR MODEL FITTING -- PARAMETERS

In Appendix B, eleven Fortran subroutines are listed that can be used to implement the iteratively reweighted least squares (IRLS) algorithms described in the body of this thesis. In Appendix A, the parameters of these subroutines are described in detail. The algorithms used by the subroutines and the uses to which they can be put, are described in the following sections of the thesis:--

GIVEN3, GIVEN2, CONV23, INITG, BSUB, VARS	Section 2.7
COMBINE	Section 3.4
COMBINEM, DIMULT, STEP	Section 3.7
NUISNC	Section 4.1

Many of the subroutines use Choleski factorizations of sums of squares matrices as parameters. These consist of an upper triangular matrix **R** and diagonal matrix **D** such that **R'DR** is the sum of squares matrix. In all subroutines except GIVEN2 and CONV23, the rows of **R** are scaled such that **R** has unit diagonal elements. The elements of **R** and **D** are then stored in "packed" format; the columns of **R** on and above the diagonal are stored in order in a linear array with the unit diagonal elements being replaced by either the inverses of the diagonal elements of **D** if these are non-zero or minus one otherwise. Subroutine GIVEN2 uses a Choleski factorization in which the diagonal elements of **R** are not one. The Choleski factorization is then stored in "unpacked"

format with the columns of **R** on and above the diagonal stored in order in a linear array and the inverses of the diagonal elements of **D** stored in a separate linear array.

SUBROUTINE GIVEN3 (R, IR, NVARS, X, V, IFAULT)

R	Real array(IR)	input: Choleski factorization in packed format output: Choleski factorization updated with new row of data or constraint
IR	Integer	input: size of R
NVARS	Integer	input: number of variables in row of data or constraint
X	Real array(NVARS)	input: new row of data or constraint output: altered
V	Real	input: variance for row (inverse of weight); zero for constraint
IFAULT	Integer	output: fault indicator, equal to -1 if constraint imposed is linearly dependent on previous constraints; R is unaltered 0 if no fault 1 if array R is too small (IR<NVARS*(NVARS+1)/2) 2 if negative variance for row 3 if inconsistent constraints are used

---

SUBROUTINE GIVEN2 (R, IR, DM, NVARS, X, V, IFAULT)

R	Real array(IR)	input: arrays R and DM are Choleski factorization in unpacked format output: arrays R and DM are Choleski factorization with new row of data or constraint
IR	Integer	input: size of R
DM	Real array(NVARS)	input: see description for R output: see description for R
NVARS	Integer	input: number of variables in row of data or constraint
X	Real array(NVARS)	input: new row of data or constraint; response is last variable output: altered
V	Real	input: variance for row (inverse of weight); zero for constraint
IFAULT	Integer	output: fault indicator, equal to -1 if constraint imposed is linearly dependent on previous constraints; R is unaltered 0 if no fault 1 if array R is too small (IR<NVARS*(NVARS+1)/2) 2 if negative variance for row 3 if inconsistent constraints are used

SUBROUTINE CONV23 (R, IR, DM, NVARS, IFAULT)

R	Real array(IR)	input: arrays R and DM are Choleski factorization in unpacked format output: Choleski factorization in packed format
IR	Integer	input: size of R
DM	Real array(NVARS)	input: see description for R
NVARS	Integer	input: number of variables in ssp matrix
IFAULT	Integer	output: fault indicator, equal to 1 if array R is too small (IR<NVARS*(NVARS+1)/2) 0 otherwise

---

SUBROUTINE INITG (R, IR, NVARS, IFAULT)

R	Real array(IR)	output: Choleski factorization of zero matrix in packed format
IR	Integer	input: size of R
NVARS	Integer	input: number of variables in zero matrix
IFAULT	Integer	output: fault indicator, equal to 1 if array R is too small (IR<NVARS*(NVARS+1)/2) 0 otherwise

---

SUBROUTINE BSUB (R, IR, IDEP, COEFF, IC, IFAULT)

R	Real array(IR)	input: Choleski factorization of sum of squares matrix in packed format
IR	Integer	input: size of R
IDEP	Integer	input: index of dependent variable
COEFF	Real array(IC)	output: estimates of the regression coefficients of IDEP'th variable on first (IDEP-1)

IC	Integer	input: size of COEFF
IFAULT	Integer	output: fault indicator, equal to <0 when (-IFAULT)identifiability constraints were needed (non-fatal) 1 if arrays R or COEFF are too small (IR<IDEP*(IDEP+1)/2 or IC<IDEP-1) 0 otherwise

---

SUBROUTINE VARS (R, IR, S, IS, NPAR, SCALE, IFAULT)

R	Real array(IR)	input: Choleski factorization of -Hessian (packed format) e.g. array R used by STEP
IR	Integer	input: size of R
S	Real array(IS)	output: inverse of -Hessian, multiplied by SCALE, (lower triangle stored by rows). Can be same array as R
IS	Integer	input: size of S
NPAR	Integer	input: number of parameters in Hessian
SCALE	Real	input: factor by which $(-\text{Hessian})^{-1}$ should be scaled. Usually 1.0 unless nuisance parameters have been factorized out.
IFAULT	Integer	output: fault indicator, equal to 1 if IR<NPAR*(NPAR+1)/2 or IS<NPAR*(NPAR+1)/2 0 otherwise

---

SUBROUTINE COMBINE (RPLUS, IR1, RMINUS, IR2, N, WORKSP, IFAULT)

RPLUS	Real array(IR1)	input: Choleski factorization for rows with variance ≥ 0 (packed format) output: Choleski factorization of combined positive and negative parts
IR1	Integer	input: size of RPLUS
RMINUS	Real array(IR2)	input: Choleski factorization for rows with variance < 0 (packed format) output: altered
IR2	Integer	input: size of RMINUS
N	Integer	input: number of variables in sums of squares matrices
WORKSP	Real array(N)	workspace: holds rows to be passed to GIVEN3
IFAULT	Integer	output: fault indicator, equal to 1 if N<2 or IR1<N(N+1)/2 or IR2<N(N+1)/2 2 if the Hessian was not negative semi-definite 0 otherwise

---

SUBROUTINE COMBINEM (S, IS, R, IR, N, DIMAX, ODMAX, WORKSP, IFAULT)

S	Real array(IS)	input: Choleski factorization for rows with variance ≥ 0 (packed format) output: Choleski factorization of combined positive and negative parts with definiteness modification (packed format)
IS	Integer	input: size of S
R	Real array(IR)	input: Choleski factorization for rows

		with variance < 0 (packed format)
		output: altered
IR	Integer	input: size of R
N	Integer	input: number of variables in sum of squares matrices
DIMAX	Real	output: maximum absolute value of diagonal elements of Hessian
ODMAX	Real	output: maximum absolute value of off-diagonal elements of Hessian
WORKSP	Real array(N)	workspace: holds rows to be passed to GIVEN3
IFAULT	Integer	output: fault indicator, equal to -1 if definiteness modification was needed to make Hessian negative definite (non-fatal) 1 if N<2 or IS<N(N+1)/2 or IR<N(N+1)/2 2 if constraints were imposed on R 0 otherwise

---

SUBROUTINE DIMULT (S, IS, NPARAM, WORKSP, IW, IFAULT)

S	Real array(IS)	input: Choleski factorization in packed format
		output: Choleski factorization of same matrix with diagonal increased by ten percent
IS	Integer	input: size of S
NPARAM	integer	input: number of parameters
WORKSP	Real array(IW)	workspace: holds rows to be passed to GIVEN3
IW	Integer	input: size of W

IFAULT	Integer	output: fault indicator, equal to 1 if NPARAM<1 or IW<NPARAM+1 or IS<(NPARAM+1)(NPARAM+2)/2 0 otherwise

---

SUBROUTINE STEP (R, IR, NPARAM, F0, BETA0, STEP0, F1, BETA1, STEP1, PACC1, DBETA, IFAULT)

R	Real array(IR)	input: Choleski factorization of derivatives as output by COMBINE, COMBINEM or NUISNC
IR	Integer	input: size of R
NPARAM	Integer	input: number of parameters
F0	Real	input: log-likelihood corresponding to BETA0
BETA0	Real array(NPARAM)	input: parameter estimates from previous iteration
STEP0	Real	input: initial stepsize - ususally one
F1	Real	output: log-likelihood corresponding to BETA1
BETA1	Real array(NPARAM)	output: parameter estimates that increase the likelihood
STEP1	Real	output: stepsize corresponding to BETA1
PACC	Real	input: criterion ρ for whether interpolation and extrapolation are used
DBETA	Real array(NPARAM)	workspace: holds changes in parameter estimates from BSUB

IFAULT	Integer	output: fault indicator, equal to 1 if NPARAM≤0 or IR<(NPARAM+1)(NPARAM+2)/2 2 if no higher likelihood could be found. Check derivatives and try another starting value. 3 if constraints are used but unit stepsize does not work 0 otherwise

This subroutine calls subroutine BSUB described above and also a real function ALLIKE(PARAMS, IP) which must be supplied by the user to evaluate the log-likelihood at the parameter values in the real array PARAMS(IP).

SUBROUTINE NUISNC (R, IR, NPARAM, DBDC, IDBDC, DC, DCDC, DIMAX, ODMAX, WORKSP, IFAULT)

R	Real array(IR)	input: Choleski factorization of derivatives with respect to some parameters output: Choleski factorization augmented by derivatives with respect to auxiliary parameter
NPARAM	Integer	input: number of parameters including auxiliary parameter
DBDC	Real array(IDBDC)	input: second derivatives with respect to main and auxiliary parameters
IDBDC	Integer	input: size of DBDC
DC	Real	input: DC and DCDC are first and second derivatives with respect to auxiliary parameters

DCDC	Real	input: see under DC
DIMAX	Real	input: maximum absolute value on diagonal of Hessian on input output: as on input, but updated with new row on Hessian
ODMAX	Real	input: maximum absolute value off diagonal of Hessian on input output: as on input, but updated with new row on Hessian
WORKSP	Real array(NPARAM)	workspace: stores rows passed to GIVEN3
IFAULT	Integer	output: fault indicator, equal to -1 if definiteness modification was needed 1 if NPARAM≦0 or IR<(NPARAM+1)(NPARAM+2)/2 or IDBDC<(NPARAM-1) 0 otherwise

APPENDIX B : FORTRAN SUBROUTINES FOR MODEL FITTING -- CODE

```
      SUBROUTINE GIVEN3(R,IR,NVARS,X,V,IFAULT)
C
C        A SLIGHT MODIFICATION OF ALGORITHM AS 164 BY
C        STIRLING IN APPL. STATIST. (1981) VOL.30, NO.2
C
C        APPLIES GIVENS TRANSFORMS TO INCLUDE A NEW ROW OF DATA
C        OR CONSTRAINT WITH 3-MULTIPLICATION METHOD
C
      REAL R(IR),X(NVARS)
      DATA ZERO/0.0/
C
C        CHECK FOR VALID PARAMETERS
C
      IFAULT=0
      IRUSED=NVARS*(NVARS+1)/2
      IF(IR.LT.IRUSED) GOTO 1001
      IF(V.LT.ZERO) GOTO 1002
      VLOCAL=V
C
C        FOR EACH ROW OF UPPER TRIANGULAR R
C
      II=0
      DO 60 I=1,NVARS
      II=II+I
      XI=X(I)
      XI2=XI*XI
      IF(XI2.EQ.ZERO) GOTO 60
      CTEMP=R(II)
      IJ=II
      IPLUS=I+1
C
C        IF ZERO WEIGHT ON ROW OF R, SIMPLE PIVOT
C
      IF(CTEMP.GE.ZERO) GOTO 20
      R(II)=VLOCAL/XI2
      IF(I.EQ.NVARS) GOTO 70
      DO 10 J=IPLUS,NVARS
      IJ=IJ+J-1
   10 R(IJ)=X(J)/XI
      RETURN
C
C        IF INFINITE WEIGHT ON ROW OF R, SIMPLE PIVOT
C
   20 IF(CTEMP.GT.ZERO) GOTO 40
      DO 30 J=IPLUS,NVARS
      IJ=IJ+J-1
   30 X(J)=X(J)-XI*R(IJ)
      GOTO 60
C
C        OTHERWISE ORDINARY GIVENS ROTATION
C
   40 VNEW=VLOCAL+CTEMP*XI2
      C=VLOCAL/VNEW
      S=CTEMP*XI/VNEW
      VLOCAL=VNEW
      R(II)=CTEMP*C
      IF(I.EQ.NVARS) GOTO 70
      DO 50 J=IPLUS,NVARS
      IJ=IJ+J-1
      RTEMP=C*R(IJ)+S*X(J)
      X(J)=X(J)-XI*R(IJ)
   50 R(IJ)=RTEMP
C
```

```
   60 CONTINUE
C
C        CHECK FOR INCONSISTENT OR DUPLICATED CONSTRAINTS
C
   70 IF(R(IRUSED).EQ.ZERO) GOTO 1003
      IF(VLOCAL.EQ.ZERO) IFAULT= -1
      RETURN
C
C        ERROR FLAG SET
C
 1003 IFAULT=IFAULT+1
 1002 IFAULT=IFAULT+1
 1001 IFAULT=IFAULT+1
      RETURN
      END
```

```
      SUBROUTINE GIVEN2(R,IR,DM,NVARS,X,V,IFAULT)
C
C        MODIFIED UNPACKED VERSION OF ALGORITHM AS 164 BY
C        STIRLING IN APPL. STATIST. (1981) VOL.30, NO.2
C
C        APPLIES GIVENS TRANSFORMS TO INCLUDE A NEW ROW OF DATA
C        OR CONSTRAINT  WITH  2-MULTIPLICATION  METHOD
C
      REAL R(IR),DM(NVARS),X(NVARS)
      LOGICAL RSCX,RSCR
C
C        MACHINE DEPENDENT DECLARATIONS TO ALLOW INTEGER ARITHMETIC ON
C        EXPONENTS OF 64-BIT REALS ON THE PRIME 750
C
      INTEGER*2 IVLOCAL(4),ICTEMP(4),IRIJ(4),IXJ(4),IXI(4),IRII(4),
     *          MAXEXP,MAXEX2
      EQUIVALENCE (IVLOCAL(1),VLOCAL),(ICTEMP(1),CTEMP),
     *          (IRIJ(1),RIJ),(IXJ(1),XJ),(IXI(1),XI),(IRII(1),RII)
      DATA MAXEXP/32638S/,MAXEX2/16319S/
C
      DATA ZERO/0.0/
C
C        CHECK FOR VALID PARAMETERS
C
      IFAULT=0
      IRUSED=NVARS*(NVARS+1)/2
      IF(IR.LT.IRUSED) GOTO 1001
      IF(V.LT.ZERO) GOTO 1002
      VLOCAL=V
C
C        FOR EACH ROW OF UPPER TRIANGULAR R
C
      II=0
      DO 60 I=1,NVARS
      II=II+I
      XI=X(I)
      IF(XI.EQ.ZERO) GOTO 60
      CTEMP=DM(I)
      IJ=II
      IPLUS=I+1
C
C        IF ZERO WEIGHT ON ROW OF R, SIMPLE PIVOT
C
      IF(CTEMP.GE.ZERO) GOTO 20
      DM(I)=VLOCAL
      DO 10 J=I,NVARS
      R(IJ)=X(J)
   10 IJ=IJ+J
      IF(I.EQ.NVARS) GOTO 70
      RETURN
C
```

```
C        IF R(II) EQUALS ZERO, SWAP ROW WITH X. (THIS WILL NEVER HAPPEN
C        IF R IS BUILT UP BY GIVEN2).
C
   20 RII=R(II)
      IF(RII.NE.ZERO) GOTO 110
      DM(I)=VLOCAL
      VLOCAL=CTEMP
      DO 120 J=I,NVARS
      TEMP=X(J)
      X(J)=R(IJ)
      R(IJ)=TEMP
  120 IJ=IJ+J
      XI=X(I)
      IJ=II
      CTEMP=DM(I)
C
C        IF INFINITE WEIGHT ON ROW OF R, SIMPLE PIVOT
C
  110 IF(CTEMP.GT.ZERO) GOTO 40
      DO 30 J=IPLUS,NVARS
      IJ=IJ+J-1
   30 X(J)=X(J)-XI*R(IJ)/RII
      GOTO 60
C
C        IF WEIGHTS COULD UNDERFLOW, RESCALE ROWS. THIS IS DONE WITH
C        INTEGER ARITHMETIC ON EXPONENTS (LAST 16 BITS OF 64-BIT REALS
C        ON PRIME) AND IS VERY MACHINE DEPENDENT. FOR MACHINES THAT
C        REPRESENT REALS DIFFERENTLY, THIS MUST BE CODED IN ASSEMBLER TO
C        TAKE ADVANTAGE OF INTEGER ARITHMETIC.
C
   40 RSCX= IVLOCAL(4).GT.MAXEXP
      IF(.NOT.RSCX) GOTO 130
      IVLOCAL(4)=IVLOCAL(4)-MAXEXP
      IXI(4)=IXI(4)-MAXEX2
  130 RSCR= ICTEMP(4).GT.MAXEXP
      IF(.NOT.RSCR) GOTO 140
      ICTEMP(4)=ICTEMP(4)-MAXEXP
      IRII(4)=IRII(4)-MAXEX2
C
C        OTHERWISE ORDINARY GIVENS ROTATION
C
  140 R1=VLOCAL/CTEMP
      R2=XI/RII
      R22=R2*R2
      IF(R22.GE.R1) GOTO 80
      CM2=1.0+R22/R1
      DM(I)=CTEMP*CM2
      R(II)=RII*CM2
      IF(I.EQ.NVARS) GOTO 70
      VLOCAL=VLOCAL*CM2
      R3=R2/R1
      DO 90 J=IPLUS,NVARS
      IJ=IJ+J-1
      XJ=X(J)
      RIJ=R(IJ)
C        MACHINE DEPENDENT SCALING OF ROWS
      IF(RSCX) IXJ(4)=IXJ(4)-MAXEX2
      IF(RSCR) IRIJ(4)=IRIJ(4)-MAXEX2
C
      X(J)=XJ-RIJ*R2
   90 R(IJ)=RIJ+XJ*R3
      GOTO 60
C
   80 SM2=1.0+R1/R22
      DM(I)=VLOCAL*SM2
      R(II)=XI*SM2
      IF(I.EQ.NVARS) GOTO 70
      VLOCAL=CTEMP*SM2
      R3=R1/R2
      DO 100 J=IPLUS,NVARS
```

```
      IJ=IJ+J-1
      XJ=X(J)
      RIJ=R(IJ)
C        MACHINE DEPENDENT SCALING OF ROWS
      IF(RSCX) IXJ(4)=IXJ(4)-MAXEX2
      IF(RSCR) IRIJ(4)=IRIJ(4)-MAXEX2
C
      X(J)=XJ/R2-RIJ
  100 R(IJ)=XJ+RIJ*R3
C
   60 CONTINUE
C
C        CHECK FOR INCONSISTENT OR DUPLICATED CONSTRAINTS
C
   70 IF(ABS(R(IRUSED)).LE.ZERO) GOTO 1003
      IF(VLOCAL.LE.ZERO) IFAULT= -1
      RETURN
C
C        ERROR FLAG SET
C
 1003 IFAULT=IFAULT+1
 1002 IFAULT=IFAULT+1
 1001 IFAULT=IFAULT+1
      RETURN
      END
```

```
      SUBROUTINE CONV23(R,IR,DM,NVARS,IFAULT)
C
C        CONVERTS FROM GIVEN2 UNPACKED FORMAT TO GIVEN3 PACKED FORMAT
C
      REAL R(IR),DM(NVARS)
C
      DATA ZERO/0.0/, ONE/1.0/
C
C        CHECK FOR VALID PARAMETERS
C
      IFAULT=1
      IRUSED=NVARS*(NVARS+1)/2
      IF(NVARS.LE.ZERO.OR.IR.LT.IRUSED) RETURN
      IFAULT=0
C
      II=0
      DO 10 I=1,NVARS
      II=II+I
      IF(DM(I).GE.ZERO) GOTO 20
      R(II)= -ONE
      GOTO 10
C
   20 RII=R(II)
      IF(RII.EQ.ZERO) GOTO 1000
      R(II)=DM(I)/(RII*RII)
      DM(I)=ONE
      IF(I.EQ.NVARS) RETURN
      IJ=II
      IPLUS=I+1
      DO 30 J=IPLUS,NVARS
      IJ=IJ+J-1
   30 R(IJ)=R(IJ)/RII
C
   10 CONTINUE
      RETURN
C
C        SET FAULT INDICATOR
C
 1000 IFAULT=2
      RETURN
      END
```

```
      SUBROUTINE INITG(R,IR,NVARS,IFAULT)
C
C        SETS R TO THE PACKED REPRESENTATION OF THE ZERO MATRIX
C
      REAL R(IR)
C
C        CHECK FOR VALID PARAMETERS
C
      IRUSED=NVARS*(NVARS+1)/2
      IFAULT=0
      IF(NVARS.LE.0.OR.IRUSED.GT.IR) GOTO 1000
      IJ=0
C
      DO 10 I=1,NVARS
      DO 15 J=1,I
      IJ=IJ+1
   15 R(IJ)=0.0
   10 R(IJ)= -1.0
      RETURN
C
 1000 IFAULT=1
      RETURN
      END
```

```
      SUBROUTINE BSUB(R,IR,IDEP,COEFF,IC,IFAULT)
C
C        ALGORITHM AS 164.1 APPL. STATIST. (1981) VOL.30, NO.2
C
C        PERFORMS BACK SUBSTITUTION TO GET PARAMETER ESTIMATES
C
      REAL R(IR),COEFF(IC)
      DATA ZERO/0.0/
C
C        CHECK FOR VALID PARAMETERS
C
      IFAULT=0
      II=IDEP*(IDEP+1)/2
      NXVARS=IDEP-1
      IF(IR.LT.II.OR.IC.LT.NXVARS) GOTO 1001
      IF(NXVARS.LT.1) RETURN
C
C        BACK SUBSTITUTION
C
      K=II
      NX=IDEP
      DO 30 I=1,NXVARS
      II=II-NX
      K=K-1
      TEMP=R(K)
      IF(R(II).LT.ZERO) IFAULT=IFAULT-1
      IF(I.EQ.1) GOTO 20
      IJ=II
      DO 10 J=NX,NXVARS
      IJ=IJ+J-1
   10 TEMP=TEMP-R(IJ)*COEFF(J)
   20 NX=NX-1
   30 COEFF(NX)=TEMP
      RETURN
C
 1001 IFAULT=1
      RETURN
      END
```

```
      SUBROUTINE VARS(R,IR,S,IS,NPAR,SCALE,IFAULT)
C
C        A MODIFIED VERSION OF ALGORITHM AS 164.3 APPL. STATIST. (1981)
C                        VOL.30, NO.2
C
C        FINDS ESTIMATE OF VAR/COVAR MATRIX OF ESTIMATES USING SCALE
C        PARAMETER PROVIDED
C
      REAL R(IR),S(IS)
      DATA ZERO/0.0/
C
C        CHECK FOR VALID PARAMETERS
C
      IFAULT=1
      IRUSED=NPAR*(NPAR+1)/2
      IF(IR.LT.IRUSED.OR.IS.LT.IRUSED) RETURN
      IFAULT=0
C
C        INVERT UNIT UPPER TRIANGULAR MATRIX
C
      IJ=0
      DO 50 I=1,NPAR
      JJ=0
      J=0
   10 J=J+1
      IJ=IJ+1
      JJ=JJ+J
      IF(J.GE.I) GOTO 50
      STEMP= -R(IJ)
      IK=IJ
      KJ=JJ
      KMAX=I-1
      KMIN=J+1
      IF(KMAX.LT.KMIN) GOTO 40
      DO 30 K=KMIN,KMAX
      IK=IK+1
      KJ=KJ+K-1
   30 STEMP=STEMP-R(IK)*S(KJ)
   40 S(IJ)=STEMP
      GOTO 10
   50 CONTINUE
C
C        APPLY IDENTIFIABILITY CONSTRAINTS
C
      II=0
      DO 60 I=1,NPAR
      II=II+I
      S(II)=SCALE*R(II)
      IF(R(II).LT.ZERO) S(II)=ZERO
   60 CONTINUE
C
C        MULTIPLY MATRICES TOGETHER TO FORM EST OF VAR
C
      II=0
      IJ=0
      DO 80 I=1,NPAR
      II=II+I
      DO 80 J=1,I
      KK=II
      IJ=IJ+1
      KI=IJ
      KJ=II
      STEMP=S(KK)
      IF(I.NE.J) STEMP=STEMP*S(IJ)
      K=I
   70 K=K+1
```

```
      IF(K.GT.NPAR) GOTO 80
      KK=KK+K
      KI=KI+K-1
      KJ=KJ+K-1
      STEMP=STEMP+S(KI)*S(KJ)*S(KK)
      GOTO 70
C
   80 S(IJ)=STEMP
      RETURN
      END
```

```
      SUBROUTINE COMBINE(RPLUS,IR1,RMINUS,IR2,N,WORKSP,IFAULT)
C
C        COMBINES THE CHOLESKI FACTORISATIONS OF OBSERVATIONS WITH
C          POSITIVE WEIGHTS (RPLUS) AND NEGATIVE WEIGHTS (RMINUS)
C        --  RETURNS ERROR CODE IF X'WX IS NOT POSITIVE DEFINITE --
C
      REAL RPLUS(IR1),RMINUS(IR2),WORKSP(N)
      REAL ZERO,V,CTEMP,VNEW,C,S
      DATA ZERO/0.0/
C
C        CHECK FOR VALID PARAMETERS
C
      IFAULT=1
      NNP1=N*(N+1)/2
      IF(N.LT.2.OR.IR1.LT.NNP1.OR.IR2.LT.NNP1) RETURN
      IFAULT=0
      NPARAM=N-1
C
      II=0
      NM1=N-1
      DO 130 I=1,NM1
C
C        CHECK FOR ZERO OR INFINITE WEIGHTS ON THE I'TH ROWS
C
      II=II+I
      IF(RMINUS(II).LT.ZERO) GOTO 130
      IF(RMINUS(II).EQ.ZERO.OR.RPLUS(II).LT.ZERO) GOTO 20
      V= -RMINUS(II)
      CTEMP= RPLUS(II)
      VNEW=V+CTEMP
      IF( VNEW.GE.ZERO) GOTO 20
C
C        ROTATE I'TH ROWS OF RMINUS AND RPLUS
C
      C=V/VNEW
      S=CTEMP/VNEW
      V= -VNEW
      RPLUS(II)=CTEMP*C
      IF(I.EQ.N) RETURN
      DO 60 J=1,I
   60 WORKSP(J)=ZERO
      IJ=II
      IPLUS=I+1
      DO 50 J=IPLUS,N
      IJ=IJ+J-1
      WORKSP(J)=RMINUS(IJ)-RPLUS(IJ)
   50 RPLUS(IJ)=C*RPLUS(IJ)+S*RMINUS(IJ)
C
C        THEN INCORPORATE REMAINDER OF I'TH ROW OF RMINUS INTO RMINUS
C
      CALL GIVEN3(RMINUS,IR2,N,WORKSP,V,IFAIL)
  130 CONTINUE
      RETURN
C
C        X'WX NOT POSITIVE DEFINITE
C
```

```
   20 IFAULT= 2
      RETURN
      END
```

```
      SUBROUTINE COMBINEM(S,IS,R,IR,N,DIMAX,ODMAX,WORKSP,IFAULT)
C
C        COMBINES THE CHOLESKI FACTORISATIONS OF OBSERVATIONS WITH
C           POSITIVE WEIGHTS (S) AND NEGATIVE WEIGHTS (R)
C        --  MAKES GILL-MURRAY DEFINITENESS MODIFICATION --
C
      REAL S(IS),R(IR),WORKSP(N),ODMAX,DIMAX
      REAL ZERO,EPS,ELEM,BETA2,SIIM,RIIM,CMAX,SIINEW,EI,A,V
      DATA ZERO/0.0/,ONE/1.0/,EPS/1.0E-8/
C
C        CHECK FOR VALID PARAMETERS
C
      IFAULT=1
      NNP1=N*(N+1)/2
      IF(N.LT.2.OR.IS.LT.NNP1.OR.IR.LT.NNP1) RETURN
      IFAULT=0
      NPARAM=N-1
C
C        APPLY CONSTRAINTS TO R AND ELIMINATE CONSTRAINED PARAMETERS
C        FROM S AND R
C
      II=0
      DO 140 I=1,NPARAM
      II=II+I
      IF(R(II).EQ.ZERO) GOTO 200
      IF(ABS(S(II)).GT.ZERO) GOTO 140
      DO 160 J=1,I
  160 WORKSP(J)=ZERO
      WORKSP(I)=ONE
      IPLUS=I+1
      IJ=II
      DO 170 J=IPLUS,N
      IJ=IJ+J-1
  170 WORKSP(J)=S(IJ)
      CALL GIVEN3(R,IR,N,WORKSP,ZERO,IFAIL)
C
      IF(I.EQ.1) GOTO 140
      JI=II
      J=I
  190 JI=JI-1
      JK=JI
      IK=II
      SJI=S(JI)
      RJI=R(JI)
      S(JI)=ZERO
      R(JI)=ZERO
      DO 180 K=IPLUS,N
      JK=JK+K-1
      IK=IK+K-1
      S(JK)=S(JK)-SJI*S(IK)
  180 R(JK)=R(JK)-RJI*R(IK)
      J=J-1
      IF(J.GE.2) GOTO 190
  140 CONTINUE
C
C        FIND MAXIMUM DIAG AND OFF-DIAG ELEMENTS
C
      ODMAX=ZERO
      DIMAX=ZERO
      IJ=NNP1-N
C
C        LOOP FOR I = N-1 DOWN TO 1
C
```

```
      I=NPARAM
C
C        LOOP FOR J = I DOWN TO 1
C
   10 J=I
      JJ=IJ
   20 IF(J.LE.0) GOTO 50
      SJJM=ZERO
      IF(S(JJ).GT.ZERO) SJJM=ONE/S(JJ)
      RJJM=ZERO
      IF(R(JJ).GT.ZERO) RJJM=ONE/R(JJ)
      IF(I.EQ.J) ELEM=SJJM-RJJM
      IF(I.NE.J) ELEM=S(IJ)*SJJM-R(IJ)*RJJM
C
C        LOOP FOR K = 1 UP TO (J-1)
C
      IF(J.EQ.1) GOTO 40
      KK=0
      IK=I*(I-1)/2
      JMINUS=J-1
      JK=J*JMINUS/2
      DO 30 K=1,JMINUS
      KK=KK+K
      IK=IK+1
      JK=JK+1
      SKKM=ZERO
      IF(S(KK).GT.ZERO) SKKM=ONE/S(KK)
      RKKM=ZERO
      IF(R(KK).GT.ZERO) RKKM=ONE/R(KK)
   30 ELEM=ELEM+S(IK)*S(JK)*SKKM-R(IK)*R(JK)*RKKM
C
   40 CONTINUE
      IF(I.EQ.J) DIMAX=AMAX1(DIMAX,ABS(ELEM))
      IF(I.NE.J) ODMAX=AMAX1(ODMAX,ABS(ELEM))
      IJ=IJ-1
      JJ=JJ-J
      J=J-1
      GOTO 20
C
   50 I=I-1
      IF(I.GT.0) GOTO 10
C
C        COMBINE POSITIVE DEFINITE AND NEGATIVE DEFINITE COMPONENTS
C
      BETA2=AMAX1(DIMAX,ODMAX/FLOAT(NPARAM),EPS)
      II=0
      DO 130 I=1,N
C
C        ALTER ITH ROW OF CHOLESKI S,  ....
C
      II=II+I
      IF(ABS(S(II)).EQ.ZERO) GOTO 130
      SIIM=ZERO
      IF(S(II).GT.ZERO) SIIM=ONE/S(II)
      IF(R(II).LT.ZERO) GOTO 130
      RIIM=ONE/R(II)
      IF(I.LT.N) GOTO 60
      TEMP=SIIM-RIIM
      S(II)= -ONE
      IF(TEMP.GT.ZERO) S(II)=ONE/TEMP
      RETURN
   60 CONTINUE
      DO 70 J=1,I
   70 WORKSP(J)=ZERO
      IJ=II
      CMAX=ZERO
      IPLUS=I+1
      DO 80 J=IPLUS,N
      IJ=IJ+J-1
      WORKSP(J)=S(IJ)
```

```
      S(IJ)=S(IJ)*SIIM-R(IJ)*RIIM
      IF(J.LT.N.AND.S(IJ).GT.CMAX) CMAX=S(IJ)
   80 CONTINUE
C
      SIINEW=AMAX1(CMAX*CMAX/BETA2,ABS(SIIM-RIIM),EPS)
      S(II)=ONE/SIINEW
C
C        ....  IF GILL-MURRAY MODIFICATION IS NEEDED UPDATE
C                                              REST OF S  ....
C
      EI=SIINEW-SIIM+RIIM
      A=SIIM/(SIIM+EI)
      IJ=II
      DO 90 J=IPLUS,N
      IJ=IJ+J-1
      R(IJ)=R(IJ)-A*WORKSP(J)
   90 S(IJ)=S(IJ)/SIINEW
      IF(EI.LE.ZERO) GOTO 100
      IFAULT= -1
      IF(EI*A.EQ.ZERO) GOTO 100
      V=ONE/(EI*A)
      CALL GIVEN3(S,IS,N,WORKSP,V,IFAIL)
C
C        ....   THEN UPDATE REST OF R
C
  100 V=SIINEW/RIIM/(SIIM+EI)
      DO 110 J=1,I
  110 WORKSP(J)=ZERO
      IJ=II
      DO 120 J=IPLUS,N
      IJ=IJ+J-1
  120 WORKSP(J)=R(IJ)
      CALL GIVEN3(R,IR,N,WORKSP,V,IFAIL)
  130 CONTINUE
C
  200 IFAULT=2
      RETURN
      END
```

```
      SUBROUTINE DIMULT(S,IS,NPARAM,WORKSP,IW)
C
C        INCREASES DIAGONAL OF HESSIAN BY TENTH
C
      REAL S(IS),WORKSP(IW)
      REAL ZERO,ONE,TEN,ELEM,V
      DATA ZERO/0.0/,ONE/1.0/,TEN/10.0/
C
C        CHECK FOR VALID PARAMETERS
C
      N=NPARAM+1
      IFAULT=1
      IF(NPARAM.LT.1.OR.IW.LT.N.OR.IS.LT.N*(N+1)/2) RETURN
      IFAULT=0
C
C        FIND DIAGONAL ELEMENT
C
      II=0
      DO 30 I=1,NPARAM
      IJ=II
      II=II+I
      IF(S(II).EQ.ZERO) GOTO 30
      ELEM=ZERO
      IF(S(II).GT.ZERO) ELEM=ONE/S(II)
      IF(I.EQ.1) GOTO 10
      JJ=0
      IMINUS=I-1
      DO 50 J=1,IMINUS
```

```
      JJ=JJ+J
      IJ=IJ+1
      IF(S(JJ).GT.ZERO) ELEM=ELEM+S(IJ)*S(IJ)/S(JJ)
   50 CONTINUE
C
C        AND ADD A TENTH OF ITS VALUE TO IT
C
   10 DO 20 J=1,N
   20 WORKSP(J)=ZERO
      WORKSP(I)=ONE
      V=TEN/ELEM
      CALL GIVEN3(S,IS,N,WORKSP,V,IFAIL)
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STEP(R,IR,NPARAM,FO,BETAO,STEPO,F1,BETA1,STEP1,
     *                            PACC,DBETA,IFAULT)
C
C        PERFORMS BARD'S STEPSIZE ALGORITHM.  AN EXTERNAL FUNCTION
C        ALLIKE IS ASSUMED TO BE PROVIDED BY THE USER TO EVALUATE
C        THE LOG-LIKELIHOOD
C
      REAL R(IR),BETAO(NPARAM),BETA1(NPARAM),DBETA(NPARAM)
      LOGICAL STEPONE
      DATA ONE/1.0/,ZERO/0.0/,HALF/0.5/,TWO/2.0/,QUART/0.25/
      DATA SMAX/1.0E3/,EPS/1.0E-13/
C
C        CHECK FOR VALID PARAMETERS
C
      IFAULT=1
      NPPLUS=NPARAM+1
      NRUSED=NPPLUS*(NPPLUS+1)/2
      IF(NPARAM.LE.0.OR.IR.LT.NRUSED.OR.PACC.LT.ZERO.OR.PACC.GT.ONE)
     *                                                    RETURN
      IFAULT=0
C
C        EVALUATE INITIAL STEP
C
      CALL BSUB(R,IR,NPPLUS,DBETA,NPARAM,IFAIL)
      GO=ZERO
      NN=NPARAM*NPPLUS/2
      STEPONE=.FALSE.
      II=0
      DO 30 I=1,NPARAM
      II=II+I
      IF(R(II).EQ.ZERO.AND.R(NN+I).NE.ZERO) STEPONE=.TRUE.
   30 IF(R(II).GT.ZERO) GO=GO+R(NN+I)*R(NN+I)/R(II)
      IF(STEPONE.AND.STEPO.NE.ONE) GOTO 50
      IF(GO.EQ.ZERO) GOTO 20
      SMIN=EPS*ABS(FO/GO)
      STEP1=STEPO
      F1=EVAL(BETAO,DBETA,STEP1,BETA1,NPARAM)
      DENOM=(STEP1*GO+FO-F1)
      SNEXT=TWO*STEP1
      IF(DENOM.NE.ZERO) SNEXT=HALF*GO*STEP1*STEP1/DENOM
C
C        IF LIKELIHOOD INCREASES, TRY INTERPOLATION OR EXTRAPOLATION
C
      IF(F1.LE.FO) GOTO 40
      IF(STEPONE) RETURN
      IF(SNEXT.LE.ZERO) SNEXT=TWO*STEP1
      SNEXT=AMIN1(SNEXT,SMAX)
      IF((ONE-PACC)*STEP1.LE.SNEXT.AND.(ONE-PACC)*SNEXT.LE.STEP1)
     *                                                    RETURN
      F2=EVAL(BETAO,DBETA,SNEXT,DBETA,NPARAM)
      IF(F2.LE.F1) RETURN
```

```
      STEP1=SNEXT
      F1=F2
      DO 10 I=1,NPARAM
   10 BETA1(I)=DBETA(I)
      RETURN
C
C        OTHERWISE, INTERPOLATE UNTIL LIKELIHOOD INCREASES
C
   40 IF(STEPONE) GOTO 50
      STEP1=AMAX1(QUART*STEP1,SNEXT)
      IF(STEP1.LE.SMIN) GOTO 20
      F1=EVAL(BETA0,DBETA,STEP1,BETA1,NPARAM)
      IF(F1.GT.F0) RETURN
      SNEXT=HALF*G0*STEP1*STEP1/(STEP1*G0+F0-F1)
      GOTO 40
C
C        IF NO BIGGER LIKELIHOOD CAN BE FOUND, ERROR
   20 IFAULT=2
      RETURN
C
C        UNIT STEP SIZE NEEDED, ERROR
C
   50 IFAULT=3
      RETURN
      END
C
C
C
      REAL FUNCTION EVAL(BETA0,DBETA,STEP,BETA1,NPARAM)
C
C        EVALUATES LOG-LIKELIHOOD FOR GIVEN STEPSIZE
C
      REAL BETA0(NPARAM),DBETA(NPARAM),BETA1(NPARAM)
      DO 10 I=1,NPARAM
   10 BETA1(I)=BETA0(I)+STEP*DBETA(I)
      EVAL=ALLIKE(BETA1,NPARAM)
      RETURN
      END
```

```
      SUBROUTINE NUISNC(R,IR,NPARAM,DBDC,IDBDC,DC,DCDC,DIMAX,ODMAX,
     *                                          WORKSP,IFAULT)
C
C        INCORPORATES DERIVATIVES WITH RESPECT TO AN AUXILIARY PARAMETER
C        INTO A CHOLESKI FACTORISATION WHICH HOLDS THE HESSIAN AND
C        GRADIENT FOR OTHER PARAMETERS  --  MAKES GILL-MURRAY DEFINITENES
S
C        MODIFICATION TO GET A NEGATIVE DEFINITE HESSIAN
C
      REAL R(IR),DBDC(IDBDC),WORKSP(NPARAM),DC,DCDC,DIMAX,ODMAX
      REAL ZERO,EPS,ONE,BETA2,W,A,TEMP,B
      DATA ZERO/0.0/,EPS/1.0E-7/,ONE/1.0/
C
C        CHECK FOR VALID PARAMETERS
C
      NPPLUS=NPARAM+1
      NPLESS=NPARAM-1
      LO=NPARAM*NPPLUS/2
      IFAULT=1
      IF(NPARAM.LE.1.OR.LO+NPPLUS.GT.IR.OR.IDBDC.LT.NPLESS) RETURN
      IFAULT=0
C
C        COPY DERIVATIVES INTO R AFTER ELEMENTS OF KDK' AND UPDATE
C        DIMAX AND ODMAX
C
      LI=LO
      DO 10 I=1,NPLESS
      LI=LI+1
```

```
      ODMAX=AMAX1(ODMAX,ABS(DBDC(I)))
   10 R(LI)= -DBDC(I)
      R(LI+1)= DC
      DIMAX=AMAX1(DIMAX,ABS(DCDC))
      BETA2=AMAX1(DIMAX,ODMAX/NPARAM,EPS)
C
C        REPLACE EACH DERIVATIVE BY THE APPROPRIATE ELEMENT FROM THE
C        REPRESENTATION OF MFM'
C
      LI=LO
      IJ=0
      DO 50 I=1,NPARAM
      LI=LI+1
      IF(I.EQ.1) GOTO 30
      IMINUS=I-1
      LJ=LO
      DO 20 J=1,IMINUS
      LJ=LJ+1
      IJ=IJ+1
   20 R(LI)=R(LI)-R(IJ)*R(LJ)
   30 IJ=IJ+1
C
C        ADD TO DIAGONAL OF HESSIAN TO MAKE IT POSITIVE DEFINITE,
C        IF NESSECARY
C
      IF(I.EQ.NPARAM.OR.ABS(R(IJ)).LE.ZERO) GOTO 50
      RIIM=ZERO
      IF(R(IJ).GT.ZERO) RIIM=ONE/R(IJ)
      W=AMAX1(R(LI)*R(LI)/BETA2, RIIM, EPS) - RIIM
      IF(W.LE.ZERO) GOTO 50
      V=ONE/W
      IFAULT= -1
      DO 40 J=1,NPARAM
   40 WORKSP(J)=ZERO
      WORKSP(I)=ONE
      CALL GIVEN3(R,IR,NPARAM,WORKSP,V,IFAULT)
   50 CONTINUE
C
C        INTERCHANGE LAST TWO ROWS AND COLS OF MFM'
C
      A= -DCDC
      LI=LO
      KI=LO-NPARAM
      II=0
      DO 60 I=1,NPLESS
      LI=LI+1
      KI=KI+1
      II=II+I
      TEMP=R(LI)
      R(LI)=R(KI)
      R(KI)=TEMP*R(II)
   60 A=A-R(KI)*TEMP
C
      KI=KI+1
      B=ONE/R(KI)
      IF(A.LT.EPS) IFAULT= -1
      R(KI)=ONE/AMAX1(ABS(A),EPS)
      LI=LI+1
      TEMP=R(LI)
      R(LI)=TEMP*R(KI)
      TEMP=B-TEMP*R(LI)
      R(LI+1)= -ONE
      IF(TEMP.GT.ZERO) R(LI+1)=ONE/TEMP
C
      RETURN
      END
```

REFERENCES

AITKEN, A.C. (1935) "On Least Squares and Linear Combination of Observations", Proceedings of the Royal Society of Edinburgh, 62, 138-146.

AITKIN, M. and CLAYTON, D. (1980), "The Fitting of Exponential, Weibull and Extreme Value Distributions to Complex Censorred Survival Data Using GLIM", Applied Statistics, 29, 156-163.

ANDERSON, J,A, and PHILIPS, P.R. (1981), "Regression, Discrimination and Measurement Models for Ordered Categorical Variables", Applied Statistics, 30, 22-31.

ANDERSON, N, and KARASALO, I. (1975), "On Computing Bounds for the Singular Values of a Triangular Matrix", BIT, 15, 1-4.

ANDREWS, D.F. (1974), "A Robust Method for Multiple Linear Regression", Technometrics, 16, 523-531.

ANDREWS, D.F., BICKEL, P.J., HAMPEL, F.R., HUBER, P.J., ROGERS, W.H. AND TUKEY, J.W. (1972), Robust Estimates of Location: Survey and Advances, Princeton: Princeton University Press.

ANDREWS, D.F. and MALLOWS, C.L. (1974), "Scale Mixtures of Normal Distributions", Journal of the Royal Statistical Society, B, 36, 99-102.

BAKER, R.J. and NELDER, J.A.(1978), General linear interactive modelling (GLIM), Release 3, Oxford: Numerical Algorithms Group.

BARD, Y. (1974), Nonlinear Parameter Estimation, New York: Academic Press.

BARHAM, R.H. and DRANE, W. (1972), "An Algorithm for Least Squares Estimation of Nonlinear Parameters When Some of the Parameters are Linear", Technometrics, 14, 757-765.

BEALL, G. (1942), "The Transformation of Data from Entomological Field Experiments so that the Analysis of Variance Becomes Applicable"", Biometrika, 32, 243-262.

BERNARDO, J.M. (1976), "Algorithm AS103: Psi (Digamma) Function", Applied Statistics, 25, 315-317.

BISHOP, Y.M.M., FIENBERG, S.E. and HOLLAND, P.W. (1975), Discrete Multivariate Analysis, Cambridge, Mass: The MIT Press.

BJORCK, A. (1967), "Solving Least Squares Problems by Gram-Schmidt Orthogonalisation", BIT, 7, 1-21.

BOX, G.E.P. and COX, D.R. (1964), "An Analysis of Transformations", Journal of the Royal Statistical Society B, 26, 211-252.

BOYLES, R.A. (1983), "On the Convergence of the EM Algorithm", Journal of the Royal Statistical Society, B, 45, 47-50.

CAUCHY, A. (1847), "Methode Generale pour la Resolution des Systems d'Equations Simultanes", Comptes Rendu de l'Academie des Science de Paris, 25, 536-538.

CHAMBERS, J.M. (1973), "Fitting Nonlinear Models: Numerical Techniques", Biometrika, 60, 1-13.

-------- (1977), Computational Methods for Data Analysis, New York: John Wiley.

CHATFIELD, C. (1969), "Discrete Distributions on Market Research", in Random Counts in Scientific Work, vol.3, ed. G.P. Patil, 163-181.

CHATFIELD, C. AND GOODHARDT, G.J. (1970), "The Beta-binomial Model for Consumer Purchasing Behaviour", Applied Statistics, 19, 240-250.

CLARKE, M.R.B. (1971), "Algorithm AS41: Updating the Sample Mean and Dispersion Marix", Applied Statistics, 20,206-209.

-------— (1981), "Algorithm AS163: A Givens Algorithm for Moving from One Linear Model to Another Without Going Back to the Data", Applied Statistics, 30, 198-203.

CLAYTON, D.G. (1983), "Fitting a General Family of Failure-time Distributions using GLIM", Applied Statistics, 32, 102-109.

COX, C.(1984), "Generalized linear models - the missing link", Applied Statistics, 33, 18-24.

COX, D.R. (1972), "Regression Models and Life Tables", Journal of the Royal Statistical Society, B, 34, 187-220.

CROWDER, M.J. (1978), "Beta-binomial Anova for Proportions", Applied Statistics, 27, 34-37.

CUNNINGHAM, E.P. and HENDERSON, C.R. (1968), "An Iterative Procedure for Estimating Fixed Effects and Variance Components in Mixed Model Situations", Biometrics, 24, 13-25.

DANIEL, C. and WOOD, F.S. (1971), Fitting Equations to Data, New York: John Wiley.

DANIEL, J.W., GRAGG, W.B., KAUFMAN, L. AND STEWART, G.W. (1976), "Reorthogonalisation and Stable Algorithms for Updating the Gram-Schmidt QR Factorisation", Mathematics of Computation, 30, 772-295.

DAVIDON, W.C. (1959), "Variable Metric Method for Minimisation", A.E.C.Research and Development Report, ANL-5990, Chicago: Argonne National Laboratory.

DEMPSTER, A.P., LAIRD, N.M. and RUBIN, D.B. (1977), "Maximum Likelihood from Incomplete Data Via the EM Algorithm", Journoal of the Royal Statistical Society, B, 39, 1-38.

-------- (1980), "Iteratively Rewieghted Least Squares for Linear Regression When Errors are Normal/Independent Distributed", Multivariate Analysis, 5, 35-57.

DEMPSTER, A.P., SELWYN, M.R., PATEL, C.M. and ROTH, A.J. (1984), "Statistical and Computational Aspects of Mixed Model Analysis", Applied Statistics, 33, 203-412.

DENNIS, J.E., GAY, D.M. and WELSCH, R.E. (1981), "An Adaptive Nonlinear Least Squares Algorithm", ACM Transactions on Mathematical Software, 7, 348-368.

EFFRON, B. and HINKLEY, D.V. (1978), "Assessing the Accuracy of the Maximum Likelihood Estimator: Observed Versus Expected Fisher Information", Biometrika, 65, 457-487.

EZEKIEL, M. and FOX, K.A. (1959), Methods of Correlation and Regression Analysis, New York: John Wiley.

FIENBERG, S.E. (1980), The Analysis of Cross-classified Categorical Data, Cambridge, Mass: The MIT Press.

FINNEY, D.J. (1944), "The Application of the Probit Method to Toxicity Test Data Adjusted for Mortality in the Controls", Annals of Applied Biology, 31, 68-74.

-------- (1947), Probit Analysis, Cambridge: Cambridge University Press.

FINNEY, D.J. and PHILLIPS, P. (1977), "The Form and Estimation of a Variance Function, with Particular Reference to Radioimmunoassay", Applied Statistics, 26, 312-320.

GENTLEMAN, W.M. (1973), "Least Squares Computations by Givens Transformations Without Square Roots", Journal of the Institute of Mathematics and its Applications, 12, 329-336.

GILL, P.E. and MURRAY, W. (1974), "Newton-type Methods for Unconstrained and Linearly Constrained Optimization", Mathematical Programming, 7, 311-350.

GLASER, R.E. (1984), "Estimation for a Weibull Accelerated Life Testing Model", Naval Research Logistics Quarterly, 31, 559-570.

GOLDFELD, S.M., QUANDT, R.E. and TROTTER, H.F. (1966), "Maximaiation by Quadratic Hill Climbing", Econometrica, 34, 541-551.

GOLUB, G.H. (1965), "Numerical Methods for Solving Linear Least Squares Problems", Numerische Mathematik, 7, 206-216.

GOLUB, G.H. and PEREYRA, V. (1973), "The Differentiation of Pseudo-inverses and Nonlinear Least Squares Problems whose Variables Separate", SIAM Journal of Numerical Analysis, 10, 413-432.

GOLUB, G.H. and REINSCH, C, (1970), "Singular Value Decompositions and Least Squares Solutions", Numerische Mathematik, 15, 403-420.

GOLUB, G. and VAN LOAN, C.F. (1983), Matrix Computations, Baltimore: John Hopkins University Press.

GREEN, P.J. (1984), "Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives", Journal of the Royal Statistical Society, B, 46, 149-192.

GREENSTADT, J. (1967), "On the Relative Efficiencies of Gradient Methods", Mathematics of Computation, 21, 360-367.

GREENWOOD, M. and YULE, G.U. (1920), "An Inquiry into the Nature of Frequency Distributions Representative of Multiple Happenings with Particular Reference to the Occurrence of Multiple Attacks of Disease of or Repeated Accidents", Journal of the Royal Statistical Society, 83, 255-279.

GRIFFITHS, D.A. (1973), "Maximum Likelihood Estimation for the Beta-binomial Distribution and an Application to the Household

Distribution of the Total Number of Cases of a Disease", Biometrics, 29, 637-648.

GUTTMAN, I., PEREYRA, V. and SCOLNIK, H.D. (1973), "Least Squares Estimation for a Class of Nonlinear Models", Technometrics, 15, 209-218.

HAMMARLING, S. (1974), "A Note on Modifications to the Givens Plane Rotation", Journal of the Institute of Mathematics and its Applications, 13, 215-218.

HARVEY, A.C. (1976), "Estimating Regression Models with Multiplicative Heteroscedasticity", Econometrica, 44, 461-465.

HARVILLE, D.A. (1973), "Fitting Partially Linear Models by Weighted Least Squares", Technometrics, 15, 509-515.

-------- (1977), "Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems", Journal of the American Statistical Association, 72, 320-338.

HESTENES, M.R. and STIEFEL, E. (1952), "Methods of Conjugate Gradients for Solving Linear Systems", Journal of Research of the National Bureau of Standards, 49, 409-436.

HILDRETH, C. and HOUCK, J.P. (1968), "Some Estimators for a Linear Model with Random Coefficients", Journal of the American Statistical Association, 72, 320-338.

HUBER, P.J. (1964), "Robust Estimation of a Location Parameter", Annals of Mathematical Statistics, 35, 73-101.

ISHII, G. and HAYAKAWA, R. (1960), "On the Compound Binomial Distribution", Annals of the Institute of Statistical Mathematics, 11, 69-80.

JENNRICH, R.I. and MOORE, R.H.(1975), "Maximum likelihood estimation by means of nonlinear least squares", Proceedings of the Statistical Computing Section of the American Statistical Association, 57-65.

JOHNSON, N.L. and KOTZ, S. (1969), Distributions in Statistics: Discrrete Distributions, New York: John Wiley.

JORGENSEN, B.(1983), "Maximum likelihood estimation and large-sample inference for generalized linear and non-linear regression models", Biometrika, 70, 19-28.

KALBFLEISCH, J.D. and PRENTICE, R.L. (1980), The Statistical Analysis

of Failure Time Data, New York: John Wiley.

KARASALO, I. (1974), "A Criterion for Truncation of the QR Decomposition Algorithm for the Singular Least Squares Problem", BIT, 14, 156-166.

KAUFMAN, L. (1975), "A Variable Projection Method for Solving Separable Nonlinear Least Squares Problems", BIT, 15, 49-57.

KENNEDY, W.J., and GENTLE, J.E. (1980), Statistical Computing, New York: Marcel Dekker.

KILPATRICK, S.J. (1977), "An Empirical Study of the Distribution of Episodes of Illness Recorded in the 1970-71 National Morbidity Survey", Applied Statistics, 26, 26-33.

KOOPMANS, T.C. and HOOD W.C. (1953), "The Estimation of Simultaneous Linear Economic Relationships", in Studies in Econometric Method, eds. W.C. Hood and T.C. Koopmans, New York: John Wiley, 112-199.

LAWTON, W.H. and SYLVESTRE, E.A. (1971), "Elimination of Linear Parameters in Nonlinear Regression", Technometrics, 13, 461-467.

LEMEIRE, F. (1975), "Bounds for Condition Numbers of Triangular and Trapezoid Matrices", BIT, 15, 58-64.

LOUIS, T.A. (1982), "Finding the Observed Information Matrix When Using the EM Algorithm", Journal of the Royal Statistical Society, B, 44, 226-233.

MANLY, B.F.J. (1976), "Exponential Data Transformations", The Statistician, 25, 37-42.

MANLY, B.F.J. and CROSBIE, E.D. (1977), "Examples of the Use of GLIM", New Zealand Statistician, 12, 26-42.

MANTEL, N. and MYERS, M. (1971), "Problems of Convergence of Maximum Likelihood Iterative Procedures in Multiparameter Situations", Journal of the American Statistical Association, 66, 484-491.

MANTON, K.G. and WOODBURY, M.A.(1981), "A variance components approach to categorical data models with heterogeneous cell populations: analysis of spacial gradients in lung cancer mortality rates in North Carolina counties", Biometrics, 37, 259-269.

MARQUARDT, D.W. (1963), "An Algorithm for Least Squares Estimation of Nonlinear Parameters", Journal of the Society for Industrial and Applied Mathematics, 11,431-441.

MARTIN, J.T. (1940), "The problem of the Evaluation of Rotenone-containing Plants. V. The Relative Toxicities of Different Species of Derris", Annals of Applied Biology, 27, 274-294.

MCCULLAGH, P. (1980), "Regression Models for Ordinal Data", Journal of the Royal Statistical Society, B, 42, 109-142.

MOORE, R.J. (1982), "Algorithm AS187 : Derivatives of the Incomplete Gamma Integral", Applied Statistics, 31, 330-335.

NATIONAL ALGORITHMS GROUP (1983), NAG Fortran Library Manual, Mark 10, Vol.3, Oxford: Numerical Algorithms Group.

NELDER, J.A. (1968), "Weighted Regression, Quantal Response Data and Inverse Polynomials", Biometrics, 24, 979-985.

-------- (1974), "Log Linear Models for Contingency Tables: A Generalisation of Classical Least Squares", Applied Statistics, 23, 323-329.

NELDER, J.A. and MEAD, R. (1965), "A Simplex Method for Function Minimisation", Computer Journal, 7, 308-313.

NELDER, J.A. and WEDDERBURN, R.W.M. (1972), "Generalized Linear Models", Journal of the Royal Statistical Society A, 135, 370-384.

PATTERSON, H.D. and THOMPSON, R. (1971), "Recovery of Inter-block Information when Block Sizes are Unequal", Biometrika, 58, 545-554.

PLACKETT, R.L. (1972), "The Discovery of the Method of Least Squares", Biometrika, 59, 239-251.

PREGIBON, D. (1980), "Goodness of Link Tests for Generalised Linear Models", Applied Statistics, 29, 15-24.

RAAB, G.M. (1981), "Estimation of a Variance Function, with Application to Immunoassay", Applied Statistics, 30, 32-40.

RICHARDS, F.S.G. (1961), "A Method of Maximum-Likelihood Estimation", Journal of the Royal Statistical Society B, 21, 469-475.

ROSS, G.J.S. (1970), "The Efficient Use of Function Minimization in Nonlinear Maximum Likelihood Estimation", Applied Statistics, 19, 205-221.

ROSS, G.J.S. (1982), "Least Squares Optimisation of General Log-Likelihood Functions and Estimation of Separable Linear Parameters", COMPSTAT 1982, Vienna: Physica-Verlag, 406-411.

RUHE, A. and WEDIN, P.A. (1980), "Algorithms for Separable Nonlinear Least Squares Problems", SIAM Review, 22, 318-337.

RUTEMILLER, H.C. and BOWERS, D.A. (1968), "Estimation in a Heteroscedastic Regression Model", Journal of the American Statistical Association, 63, 552-227.

SCALLON, A., GILCHRIST, R. and GREEN, M. (1984), "Fitting Parametric Link Functions in Generalized Linear Models", Computational Statistics and Data Analysis, 2, 37-49.

SCHNEIDER, B.E. (1978), "Algorithm AS121: Trigamma Function", Applied Statistics, 27, 97-99.

SEAL, H.L. (1967), "The Historical Development of the Gauss Linear Model", Biometrika, 54, 1-24.

SKELLAM, J.G. (1948), "A Probability Distribution Derived from the Binmomial Distribution by Regarding the Probability of Success as Variable Between the Sets of Trials", Journal of the Royal Statistical Society, B, 10, 257-261.

SPITZER, J.J. (1982), "A Fast and Efficient Algorithm for the Estimation of Parameters in Models with the Box-and-Cox Transformation", Journal of the American Statistical Association, 77, 760-766.

STIRLING, W.D. (1981) "Algorithm AS164: Least Squares Subject to Linear Constraints", Applied Statistics, 30, 204-212.

-------- (1984), "Iteratively Reweighted Least Squares for Models with a Linear Part", Applied Statistics, 33, 7-17.

------- (1985), "Heteroscedastic models and an application to block designs", Applied Statistics, 34, ???-???.

------- (19??), "Algorithm AS???: Fitting models with a linear part and nuisance parameters", Applied Statistics, ??, ???-???.

SWALLOW, W.H. and MONAHAN, J.F. (1984), "Monte Carlo Comparisons of ANOVA, MIVQUE, REML and ML Estimators of Variance Components", Technometrics, 26, 47-57.

THOMPSON, R. and BAKER, R.J.(1981), "Composite link functions in generalized linear models", Applied Statistics, 30, 125-131.

WAMPLER, R.H. (1970), "A Report on the Accuracy of Some Widely Used Least Squares Computer Programs", Journal of the American Statistical Association, 65, 549-565.

WEDDERBURN, R.W.M. (1974a), "Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method", Biometrika, 61, 439-447.

-------- (1974b), "Generalized Linear Models Specified in Terms of Constraints", Journal of the Royal Statistical Society, B, 36, 449-454.

WHITE, G.C. and EBERHARDT, L.E. (1980). "Statistical Analysis of Deer and Elk Pellet-group Data", Journal of Wildlife Management, 44, 121-131.

WILKINSON, J.H. (1977), "Some Recent Advances in Numerical Linear Algebra", in The State of the Art in Numerical Analysis, ed. D.Jacobs, London: Academic Press, 3-53.

WILLIAMS, E.J. (1959), Regression Analysis, New York: John Wiley.