This work was supported by a research grant (NGR 34-003-040) from the National Aeronautics and Space Administration, and by a Public Health Service Research Career Development Award (1-K3-GM-31,732) and research grant (GM HD 16697) from the National Institute of General Medical Sciences.

A SUBROUTINE PACKAGE FOR MAXIMUM LIKELIHOOD ESTIMATION (MAXLIK)

by

Ellen B. Kaplan and R. C. Elston

Department of Biostatistics
School of Public Health
University of North Carolina

Institute of Statistics Mimeo Series No. 823

JUNE 1972

(Revised, March 1978)

Acknowledgements

The authors wish to thank Mrs. Katie C. Yelverton for her many helpful suggestions and her assistance in preparing the write-up, Mr. Philip P. Green for drafting the summary of user supplied information, Dr. Thomas G. Donnelly for his suggestions with regard to numerical differentiation, and Mrs. Delores Gold for typing. They are also grateful to Dr. M. Anne Campbell and Mr. Kelvin K. Lee, who kindly read and criticized a draft of the write-up. The matrix multiplication and inversion subroutines used are from the Packed Matrix Subroutine Library developed by the Department of Biostatistics at U.N.C.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

A Subroutine Package for Maximum Likelihood Estimation

I. IDENTIFICATION

Authors: Ellen B. Kaplan and R. C. Elston
Source Language: ANSI Standard Fortran IV

II. DESCRIPTION AND PURPOSE

Given a set of data and a statistical model, the log likelihood of the data, under that model, can be formulated. In conjunction with a user supplied subroutine which computes such a log likelihood for a given set of parameter values, this subroutine package searches the likelihood surface to find maximum likelihood estimates of the parameters and computes an estimate of their variance-covariance matrix. A set of initial values for the parameters must be supplied. If desired, the user may specify functional relationships among the parameters; and a subset of the parameters may be held fixed, and not estimated. Furthermore, upper and/or lower bounds for the parameters may be specified; it is not necessary for the log likelihood function to be defined at all points within the specified bounds, provided the user supplied subroutine for calculating it tests for undefined points. A search can be performed to verify that the estimates correspond to a point on the likelihood surface that is higher than a set of neighboring points, but only one such "local maximum" is found for each set of initial parameter values.

The subroutine package can also be used for general function maximization (or for minimization by maximizing the negative function); the resulting variance-covariance matrix is only meaningful when the function maximized is a log likelihood.

III. METHOD

A. Notation

t = total number of parameters in the model.

r = number of parameters to be estimated in the run; i.e., $t-r$ parameters are held fixed; $r \leq t$.

s = number of parameters that are functionally dependent on other parameters in the run; $s < r$.

$\underset{t\times 1}{\underline{\theta}}$ = vector whose elements are the t parameters.

$L(\theta_1,\theta_2,\ldots,\theta_t)$ = natural logarithm of the likelihood given $\underline{\theta} = (\theta_1,\theta_2,\ldots,\theta_t)$.

$\theta_{i\ min}$ = lower bound for the i-th parameter.

$\theta_{i\ max}$ = upper bound for the i-th parameter.

ε = requested maximum relative change in the estimates between the final two iterations. (For the precise interpretation of ε, see p. III-4)

δ = initial step size used in the search of the log likelihood surface.

δ_i = step size for the parameter θ_i in the current iteration.

B. Algorithm for the search of the likelihood surface

1. Initial step size for each parameter

$$\left.\begin{array}{l}\text{If } \delta=0, \text{ set } \delta_i=0.5 \\ \text{If } \delta>0, \text{ set } \delta_i=\delta\end{array}\right\} \text{ for } i=1,2,\ldots,t$$

If $\delta<0$, read in the vector $\underline{\delta_i}$ which gives an individual step size for each parameter, $i=1,2,\ldots,t$. Step sizes for parameters that are fixed at a constant value or are functionally dependent are set or read in with the others but are ignored.

2. Test for closeness to a bound

A parameter estimate θ_i is considered to be close to a bound, $\theta_{ib} = \theta_{i\,max}$ or $\theta_{i\,min}$, if the following conditions are satisfied:

either

$$|\theta_i| \geqq \varepsilon \text{ and } 1-\varepsilon \leqq \frac{\theta_{ib}}{\theta_i} \leqq 1+\varepsilon$$

or

$$|\theta_i| < \varepsilon \text{ and } |\theta_i - \theta_{ib}| \leqq \varepsilon^2 .$$

If θ_i is found to be close to a bound and, in addition to this, $\delta_i \leqq \varepsilon_{ib}$, where

$$\varepsilon_{ib} = \begin{cases} \varepsilon & \text{if } |\theta_i| > \frac{1}{2} \\ \varepsilon|\theta_i| & \text{if } \varepsilon \leqq |\theta_i| \leqq \frac{1}{2} \\ \varepsilon^2 & \text{if } |\theta_i| < \varepsilon , \end{cases}$$

then θ_i is held fixed until the basic iteration process has converged for all parameters (as described below, page III-4), and an indication that this has been done is printed. The parameter is fixed at its last value or at the bound, whichever yields the larger log likelihood.

3. Basic iteration process

The following steps *a-e* inclusive are repeated sequentially for $i=1,2,\ldots,r-s$, and this sequence comprises one basic iteration. The θ_i are considered to be renumbered so that $i=1,2,\ldots,r-s$ corresponds to the $r-s$ parameters that are not fixed or dependent. A parameter θ_i will be held fixed at its current value until the basic iteration process has converged (as described below, p. III-4), if at any time the corresponding step size δ_i becomes smaller than ε_i, where

$$\varepsilon_i = \begin{cases} \varepsilon/8 & \text{if} \quad |\theta_i| > \frac{1}{2} \\ \varepsilon|\theta_i|/8 & \text{if} \quad \varepsilon < |\theta_i| \leq \frac{1}{2} \\ \varepsilon^2/8 & \text{if} \quad |\theta_i| \leq \varepsilon. \end{cases}$$

The five basic iteration steps are:

a. A test for closeness to a bound, as in B above, is performed on the parameter estimate θ_i.

b. The log likelihood is then evaluated using three different trial values of θ_i, holding the other parameters constant. If the estimate θ_i is not close to a bound, the values used are θ_i, $\theta_i+\Delta\theta_i$, and $\theta_i-\Delta\theta_i$, where

$$\Delta\theta_i = \begin{cases} \theta_i\delta_i & \text{if} \quad |\theta_i| > \frac{1}{2} \\ \delta_i & \text{if} \quad |\theta_i| \leq \frac{1}{2} \end{cases}$$

If θ_i is close to $\theta_{i\ max}$, the three values used are $\theta_{i\ max}$, $\theta_{i\ max} - |\Delta\theta_i|/2$, and $\theta_{i\ max} - |\Delta\theta_i|$. Similarly, if θ_i is close to $\theta_{i\ min}$, the three values are $\theta_{i\ min}$, $\theta_{i\ min} + |\Delta\theta_i|/2$, and $\theta_{i\ min} + |\Delta\theta_i|$.

c. If any of these trial values of θ_i are outside of their bounds, or if they cause any functionally dependent parameters to be outside of their bounds, δ_i is replaced by $\delta_i/2$ and steps *a* and *b* are repeated until all values are within bounds.

d. Let $L(\underline{\theta}, \theta_i+\Delta\theta_i) = L(\theta_1,\ldots,\theta_{i-1}, \theta_i+\Delta\theta_i, \theta_{i+1},\ldots,\theta_t)$ and $L(\underline{\theta}, \theta_i-\Delta\theta_i) = L(\theta_1,\ldots,\theta_{i-1}, \theta_i-\Delta\theta_i, \theta_{i+1},\ldots,\theta_t)$. (In the case of the value of θ_i being close to a bound, θ_i is replaced by $\theta_{i\ max} - |\Delta\theta_i|/2$ or $\theta_{i\ min} + |\Delta\theta_i|/2$ and $\Delta\theta_i$ is replaced by $|\Delta\theta_i|/2$.) Both $L(\underline{\theta}, \theta_i+\Delta\theta_i)$ and $L(\underline{\theta}, \theta_i-\Delta\theta_i)$ are computed. If either of these values is determined to be undefined in the user supplied subroutine for the log likelihood, δ_i is replaced by $\delta_i/2$

and this step is repeated until defined values are obtained.

e. (i) If $\max[L(\underline{\theta},\theta_i-\Delta\theta_i),L(\underline{\theta}),L(\underline{\theta},\theta_i+\Delta\theta_i)] = L(\underline{\theta})$, let

$$\theta_i^* = \theta_i - \frac{[L(\underline{\theta},\theta_i+\Delta\theta_i)-L(\underline{\theta},\theta_i-\Delta\theta_i)]\Delta\theta_i}{2[L(\underline{\theta},\theta_i-\Delta\theta_i)-2L(\underline{\theta})+L(\underline{\theta},\theta_i+\Delta\theta_i)]}$$

and compute $L(\underline{\theta},\theta_i^*)$. This would be the maximum if the log likelihood were a parabola through the points $L(\underline{\theta},\theta_i-\Delta\theta_i)$, $L(\underline{\theta})$, and $L(\underline{\theta},\theta_i+\Delta\theta_i)$.

If $L(\underline{\theta},\theta_i^*) \geq L(\underline{\theta})$, replace θ_i by θ_i^* and δ_i by $\delta_i/2$ and continue with step *a* for the next parameter. If $L(\underline{\theta},\theta_i^*) < L(\underline{\theta})$, replace δ_i as follows:

$$\delta_i = \begin{cases} |\theta_i-\theta_i^*|/|\theta_i| & \text{if} \quad |\theta_i| > \tfrac{1}{2} \\ |\theta_i-\theta_i^*| & \text{if} \quad |\theta_i| \leq \tfrac{1}{2} \end{cases}$$

and go back to step *b* using the new value of δ_i.

(ii) If $\max[L(\underline{\theta},\theta_i-\Delta\theta_i),L(\underline{\theta}),L(\underline{\theta},\theta_i+\Delta\theta_i)] = L(\underline{\theta},\theta_i+\Delta\theta_i)$, θ_i is replaced by $\theta_i+\Delta\theta_i$ and δ_i is replaced by $\min(2\delta_i,0.5)$, and the program continues with step *a* for the next parameter.

(iii) If $\max[L(\underline{\theta},\theta_i-\Delta\theta_i),L(\underline{\theta}),L(\underline{\theta},\theta_i+\Delta\theta_i)] = L(\underline{\theta},\theta_i-\Delta\theta_i)$, θ_i is replaced by $\theta_i-\Delta\theta_i$ and δ_i is replaced by $\min(2\delta_i,0.5)$, and the program continues with step *a* for the next parameter.

4. Test for convergence

After completing each iteration as described in section 3 over all parameters, a test for convergence is performed. The first part of the test determines whether $\delta_i \leq \varepsilon_i$, for all i, where

$$\varepsilon_i = \begin{cases} \varepsilon & \text{if} \quad |\theta_i| > \tfrac{1}{2} \\ \varepsilon|\theta_i| & \text{if} \quad \varepsilon \leq |\theta_i| < \tfrac{1}{2} \\ \varepsilon^2 & \text{if} \quad |\theta_i| < \varepsilon \quad . \end{cases}$$

Passing this first test implies that, denoting by θ_{0i} and θ_{1i} the last two values of θ_i,

$$|\theta_{0i}-\theta_{1i}| \leq \varepsilon^2 \quad \text{if} \quad |\theta_{0i}| < \varepsilon$$

$$\text{or} \qquad 1-\varepsilon \leq \frac{\theta_{1i}}{\theta_{0i}} \leq 1+\varepsilon \quad \text{if} \quad |\theta_{0i}| \geq \varepsilon.$$

For the dependent parameters, and in those cases where any other methods are used (e.g., Newton-Raphson), this implied test, rather than the above test, is applied directly. However, if the maximum absolute difference between any of the log likelihoods calculated during a basic iteration and that found at the end of the previous iteration is less than ε^2, 10ε is used instead of ε in the above test. Also, it is possible for the user to supply a constant κ such that this test will be considered as passed if the same maximum absolute difference in log likelihoods is ever less than κ.

If this first test is not passed, the program continues with 5 below; otherwise the program continues with the second part of the test for convergence, in which an (r-s)-dimensional search is made for a larger log likelihood in the neighborhood of the current estimates θ_{1i}. Allowing any parameters that have been temporarily fixed to vary again, the 2(r-s) values of the log likelihood $L(\underline{\theta}_1, \theta_{1i} \pm \Delta\theta_{1i})$ are computed (omitting any for which the parameter point is outside permissable bounds or for which the log likelihood is not defined). For this computation $\Delta\theta_{1i}$ is defined as:

$$\Delta\theta_{1i} = \begin{cases} \varepsilon^2 & \text{if} \quad |\theta_{1i}| < \varepsilon \\ \varepsilon|\theta_{1i}| & \text{if} \quad |\theta_{1i}| \geq \varepsilon. \end{cases}$$

Let $L(\underline{\theta}_2)$ be the largest value of the log likelihood found in this search. If this is larger than $L(\underline{\theta}_1)$, and $L(\underline{\theta}_2)-L(\underline{\theta}_1) > \max(\varepsilon^2,\kappa)$, the δ_i are all set equal to ε and the basic iteration process is resumed. Otherwise the basic iteration process has converged, and the program either proceeds with a 2^{r-s}-dimensional search around the estimates (as described below, p. III-7), or, if the user elects to omit this, considers the current estimates to be the final ones.

5. Supplementary iteration process

Let $\underline{\theta}_1$ denote the current value of $\underline{\theta}$ and $\underline{\theta}_0$ denote the value of $\underline{\theta}$ at the beginning of the basic iteration. The sequence

$$2^j\underline{\theta}_1-(2^j-1)\underline{\theta}_0 = \underline{\theta}^*_j \qquad j=1,2,3,\ldots$$

is computed. (The formula applies to the independent parameters only; the dependent parameters in $\underline{\theta}^*_j$ are computed from the independent parameters in it.) The sequence is continued until either (1) $\underline{\theta}^*_j$ is outside bounds or $L(\underline{\theta}^*_j)$ is undefined or (2) $L(\underline{\theta}^*_j)$ is no longer increasing.

At that point $\underline{\theta}_1$ is replaced by the previous $\underline{\theta}^*_j$, (i.e., the $\underline{\theta}^*_j$ giving the largest defined likelihood), and this becomes the $\underline{\theta}_0$ for the beginning of the next basic iteration.

Another sequence of basic and supplementary iterations is then performed, provided the maximum number of iterations specified by the user is not exceeded. If that number is exceeded, the current results are printed and no further computations are done unless the user specifies that Newton-Raphson iteration is to be tried, starting with the current estimates.

6. 2^{r-s}-dimensional search around the estimates

When the (r-s)-dimensional search no longer gives improved values, an analogous 2^{r-s}-dimensional search is optionally made. The 2^{r-s} values of the log likelihood $L(\theta_1 \pm \Delta\theta_1, \theta_2 \pm \Delta\theta_2, \ldots, \theta_{r-s} \pm \Delta\theta_{r-s})$ are computed. If significantly improved estimates θ_i^* are found (i.e., $L(\underline{\theta}^*)-L(\underline{\theta}) > \max(\varepsilon^2, \kappa)$), the θ_i are replaced by $2\theta_i^* - \theta_i$ or by θ_i^*, whichever gives the largest log likelihood; and the basic iteration process is again resumed after setting all δ_i equal to ε.

A second 2^{r-s}-dimensional search will be performed, if necessary, but not a third; instead a warning is printed that the final estimates may not be at a maximum, and that this search algorithm may not be suitable for finding maximum likelihood estimates.

C. Variance-covariance matrix of the estimates

The computation of the variance-covariance matrix V is optional and will be omitted automatically if the iterative search of the log likelihood surface does not converge. Any parameter θ_i that is close to a bound (see p. III-2) is now fixed there regardless of the size of δ_i.

Let k ($\leq$r-s) be the number of parameters that are now neither fixed nor functionally dependent on other parameters. The k×k matrix W of numerical second partial derivatives of L with respect to these parameters is first computed as follows. (The θ_i are again considered to be renumbered so that these parameters correspond to i=1,2,...,k).

To obtain the (i,j)-th element of W, compute (unspecified arguments in L are left unchanged)

$$w_{ij} = \begin{cases} \dfrac{L(\theta_i+\Delta\theta_i,\theta_j+\Delta\theta_j)-L(\theta_i-\Delta\theta_i,\theta_j+\Delta\theta_j)-L(\theta_i+\Delta\theta_i,\theta_j-\Delta\theta_j)+L(\theta_i-\Delta\theta_i,\theta_j-\Delta\theta_j)}{4\Delta\theta_i\Delta\theta_j} & \text{if } i \neq j \\[2ex] \dfrac{L(\theta_j+\Delta\theta_j)-2L(\theta_j)+L(\theta_j-\Delta\theta_j)}{(\Delta\theta_j)^2} & \text{if } i=j, \end{cases}$$

where

$$\Delta\theta_i = \begin{cases} \varepsilon\theta_i & \text{if} \quad |\theta_i| > \tfrac{1}{2} \\ \varepsilon & \text{if} \quad |\theta_i| \leq \tfrac{1}{2} \end{cases} .$$

If necessary, however, $\Delta\theta_i$ is adjusted so that all the parameter points involved in these computations are within permissible bounds and all the values of L are defined. This is done by halving certain $\Delta\theta_i$, the halving continuing as long as necessary provided $\Delta\theta_i$ does not become less than $\varepsilon^2/8$. Should this occur, a warning is printed that the derivative cannot be evaluated at this point.

Next w_2 is calculated in the same way as w_1 with $\Delta\theta_i/2$ substituted for $\Delta\theta_i$ and $\Delta\theta_j/2$ substituted for $\Delta\theta_j$. Similarly w_3 is calculated by again halving $\Delta\theta_i$ and $\Delta\theta_j$. The (i,j)-th element of W is then obtained using the following three steps iteratively:

1. If $|w_1|, |w_2|$, and $|w_3|$ are all $<10^{-8}$, $w_{ij} = 0$.
2. If not, then let

$$R = \frac{2w_2 - w_1}{2w_3 - w_2} .$$

   If $1-\varepsilon \leq R \leq 1+\varepsilon$, $w_{ij} = (w_1 - 6w_2 + 8w_3)/3$.
3. If not, w_4 is calculated analogously to w_3, again halving $\Delta\theta_i$ and $\Delta\theta_j$. After setting $w_1 = w_2$, $w_2 = w_3$, and $w_3 = w_4$, the program returns to step 1.

These three steps are repeated as long as R is approaching unity. If at any iteration R is further away from unity than in the previous iteration, then w_{ij}

is set equal to the w_1 of the previous iteration, and possible round-off error is indicated for this element.

The matrix $-W$ is inverted, using Gauss-Jordan elimination, to obtain the variance-covariance matrix of the independent parameters that are not fixed. Let

$$\underset{k\times k}{V} = (v_{ij}) = (-W)^{-1}$$

If any parameters have converged to a bound, rows and columns of 0's are inserted in the matrix V in the positions corresponding to these parameters, so that the dimensions of V are now $(r-s)\times(r-s)$. These rows and columns are not inserted for parameters which are declared fixed by the user at the beginning of the run.

If there are functional relationships among the parameters, the variances and covariances for the dependent parameters are computed as follows and inserted in V in the positions corresponding to these parameters; it is assumed that each dependent parameter is expressed explicitly as a function of the independent parameters only. Let the s specified functions be $f_1, f_2, \ldots, f_s$. For $k-1,2, \ldots, s$ and $h=1,2, \ldots, r-s$, calculate

$$u_1 = \frac{f_k(\theta_h+\Delta\theta_h)-f_k(\theta_h-\Delta\theta_h)}{2\Delta\theta_h}$$ where the $r-s-1$ unspecified arguments of f_k are θ_i,

and similarly u_2 and u_3 substituting $\Delta\theta_h/2$ and $\Delta\theta_h/4$ respectively for $\Delta\theta_h$. Then the first derivatives, for each k and h, are obtained numerically from the sequence $u_1, u_2, u_3, \ldots$, in the same manner that the elements of W are obtained from the sequence $w_1, w_2, w_3, \ldots$.

Using these numerical estimates, the covariance between the dependent parameter θ_p and the independent parameter θ_j is taken to be

$$\sum_{i=1}^{r-s} \frac{\partial f_p}{\partial \theta_i} v_{ij};$$

the variance of θ_p is taken to be

$$\sum_{i=1}^{r-s} \sum_{j=1}^{r-s} \frac{\partial f_p}{\partial \theta_i} \frac{\partial f_p}{\partial \theta_j} v_{ij};$$

and the covariance of two dependent parameters θ_p and θ_q is taken to be

$$\sum_{i=1}^{r-s} \sum_{j=1}^{r-s} \frac{\partial f_p}{\partial \theta_i} \frac{\partial f_q}{\partial \theta_j} v_{ij}.$$

The final matrix V has dimensions r×r. Standard errors of the estimates of the parameters are computed as the square roots of the diagonal elements of V.

D. Newton-Raphson iteration

If the variance-covariance matrix is computed, one iteration of the Newton-Raphson method, using numerical derivatives, is automatically performed. The numerical first partial derivatives of the log likelihood function are computed in the manner just described for obtaining $\partial f_k/\partial \theta_h$. If, after this iteration, a parameter θ_i is outside its permissible bounds, it is replaced by the value of the appropriate bound. Next the test for convergence (p. III-4) is applied and the new log likelihood $L(\underline{\theta}_{i+1})$ compared with the previous one $L(\underline{\theta}_i)$. There are three possible outcomes:

1. If the test for convergence is passed, the program prints a message indicating that the estimates have been verified by the Newton-Raphson method. If $L(\underline{\theta}_{i+1}) > L(\underline{\theta}_i)$, $\underline{\theta}_{i+1}$ is given as the final vector of estimates, but the variance-covariance matrix is not recomputed.

2. If the test for convergence is not passed and $L(\underline{\theta}_{i+1}) < L(\underline{\theta}_i)$ or if $L(\underline{\theta}_{i+1})$ is not defined, the results of this iteration are ignored and $\underline{\theta}_i$ is the final vector of estimates.

3. If the test for convergence is not passed and $L(\underline{\theta}_{i+1}) > L(\underline{\theta}_i)$, the program proceeds with further iterations of the Newton-Raphson method (using numerical derivatives) without, however, recalculating the variance-covariance matrix at each iteration. This continues until one of the following conditions is met:

a. Convergence is obtained.

b. The log likelihood decreases.

c. The maximum number of iterations is exceeded.

If improved final estimates are obtained in this way, the variance-covariance matrix can be recomputed using these estimates.

The Newton-Raphson method can also be used by itself, either with or without recomputing the variance-covariance matrix at each iteration.

E. Possible failures

1. Possible causes of poor convergence

In practice this method has converged efficiently for a variety of models. However, in some cases the method may converge very slowly or conceivably not at all.

The most common cause of slow convergence is an almost flat likelihood surface. To overcome this, the requested maximum relative change in the estimates between the final two iterations, ε, can be increased.

Non-convergence may also be caused by oscillation among the parameters, i.e., there are two or more relative maxima which are actually equivalent and the estimates go back and forth among them. This can usually be avoided by putting appropriate bounds or restrictions on the parameters to keep them from "changing places". A smaller initial step size can also help in this situation.

2. Possible sources of error in the variance-covariance matrix

a. Round-off errors in the second derivatives; these should normally be detected, and a warning message printed, together with the values of w_1, w_2, and w_3 for the previous iteration. (See p. IV-35)

b. Errors in the inversion; the matrix (-W)(V) is computed and printed for comparison with the identity matrix.

c. Singularity; if the matrix W is singular a message is printed and the variance-covariance matrix is not printed. (In this case it is frequently possible to compute a variance-covariance matrix by specifying additional parameters as functionally dependent or fixed.)

d. Negative elements on the diagonal; this may be due to (1) round-off error in the computation of the second derivatives or of the inverse; (2) the final estimates not being near a local maximum;* this is still possible but very unlikely if the 2^{r-s}-dimensional search around the final estimates has been performed; (3) the matrix being close to singular. Whenever a diagonal element is negative, the corresponding standard error is not computed; the element of V itself is printed instead.

e. An error indication to the effect that a derivative could not be computed may occur if the likelihood is not continuous in the neighborhood of the estimates. (see p. IV-35)

It is especially important that double precision should be used in the computation of V.

* A smaller value of ε will sometimes rectify this.

IV. STRUCTURE OF THE PROGRAM

A. Basic description

The maximum likelihood subroutine package MAXLIK consists of three main parts:

1. A control subroutine which provides input of program parameters, output of results, selection of options, and a framework for using more than one method of computing the estimates, if desired.

2. Iterative algorithms for computing the maximum likelihood estimates.

3. Subroutines for computing the estimated variance-covariance matrix of the estimates.

MAXLIK is called from the user's driver program by a single subroutine call. Information concerning the analysis to be performed is supplied to MAXLIK in three ways:

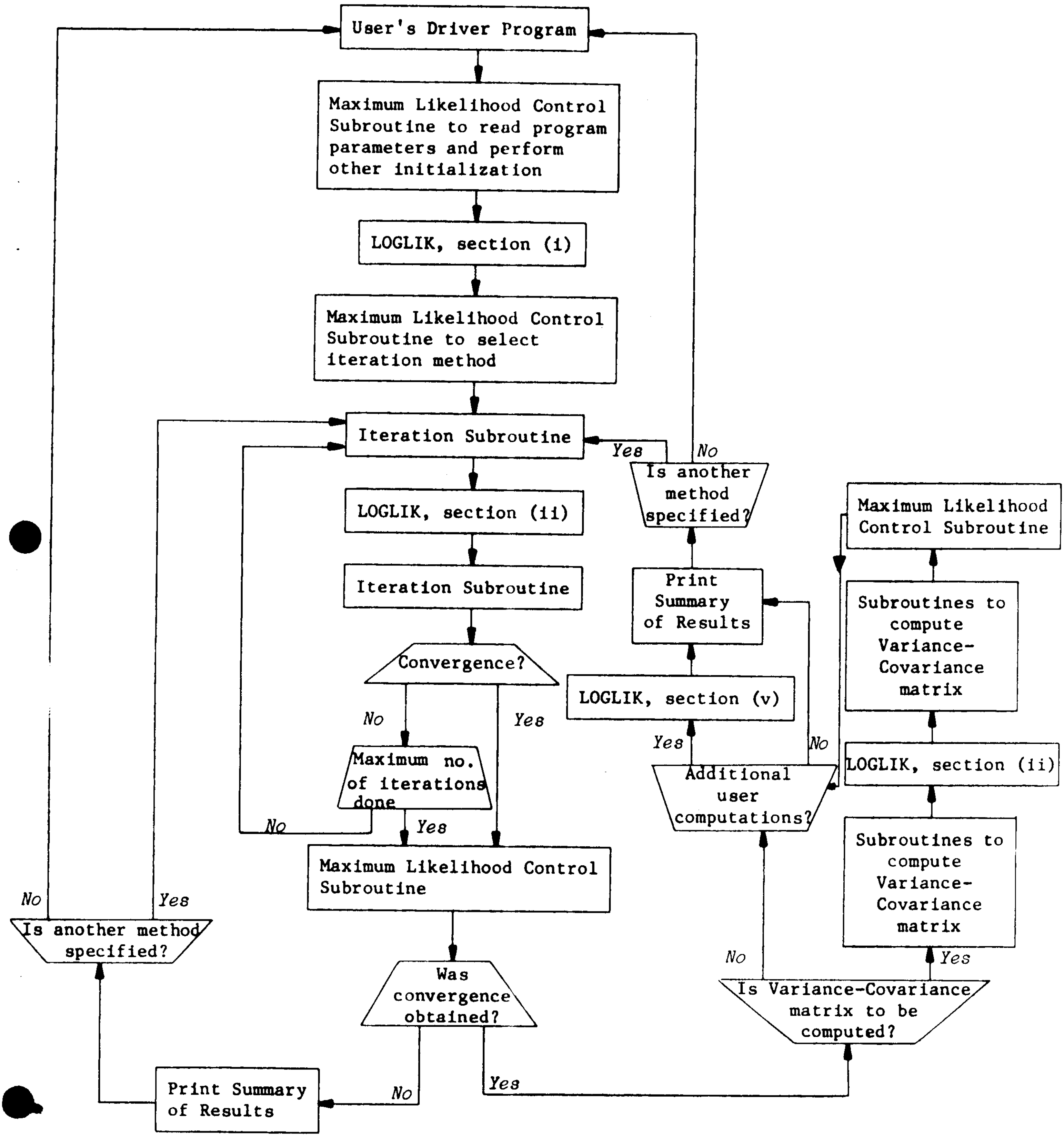
1. The subroutine call arguments. (See p. IV-3)

2. Two subroutines called by MAXLIK, LOGLIK and DEPAR, describing the model. (See p. IV-3.)

3. A set of program parameter cards read in by MAXLIK. (See p. IV-15.)

Flow chart of major steps

The flow among the various parts of the package and the user supplied driver, or calling, program and subroutines is illustrated in the flow chart on the next page.



NOTE: DEPAR is called from all subroutines as needed.

B. Driver program

1. Calling sequence

The driver, or calling, program which calls MAXLIK must contain the following statements or their equivalent.

```
DOUBLE PRECISION X,V    (if double precision is used)
DIMENSION X(m), V(u²)
EXTERNAL LOGLIK, DEPAR
CALL MAXLIK (LOGLIK,DEPAR,X,V)
```

Program variables

X   Real (double precision) array. This array contains the final estimates of the parameters after the return from the subroutine MAXLIK. It can, but does not need to, contain initial estimates for the analysis when entering MAXLIK.

V   Real (double precision) array. This array contains the variance-covariance matrix of the estimates of the independent parameters after return from MAXLIK, provided it has been computed. The matrix is stored in packed rectaugular form.

LOGLIK   Dummy subroutine name. LOGLIK refers to the user supplied subroutine which computes the log likelihood of the model. The user must replace all references to LOGLIK in the above calling program by a subroutine name of his choice.

DEPAR   Dummy subroutine name. DEPAR refers to the user supplied subroutine to evaluate functionally dependent parameters, if any, in the model. The user must replace all references to DEPAR in the above program by a subroutine name of his choice. If there are no functional relationships, the references to DEPAR may be left as they are.

The form of these subroutines is described in detail below.

The dimensions m and u^2 listed in the driver program must be such that

m $\geq$ the maximum number of parameters to be utilized by the particular model specified in LOGLIK ($\geq$NT, parameter card 2, col. 37-39).

u $\geq$ the maximum number of parameters estimated (i.e. not held fixed) by the particular model specified in LOGLIK, for which the variance-covariance matrix of the estimates is to be computed. ($\geq$NR, parameter card 2, col. 40-42).

In the driver program m and u^2 must be replaced by the appropriate integer constants.

The names replacing LOGLIK and DEPAR may be ENTRY names rather than SUBROUTINE names if the Fortran compiler used has the ENTRY statement.

The results obtained in MAXLIK can be used for further computations either in the driver program or in the subroutine LOGLIK (p. IV-5). Results other than X and V as well as exit codes will be available in the driver program if the COMMON statements described on p. IV-6, p. IV-13, IV-14 are included here.

2. Multiple calls

The following definitions will aid the discussion of the use of MAXLIK:

A <u>method</u> denotes the computations done to obtain the maximum likelihood estimates using one of the iterative techniques included in MAXLIK.

An <u>analysis</u> denotes the computations done to get the estimates for one model and one set of data using one or more methods.

A <u>run</u> consists of one or more analyses done during one computer job.

Several methods can be used during an analysis by setting the appropriate parameters on parameter card 2 (p. IV-21). The user can specify either that a new method be tried only if the previous method did not give convergence, or that all methods listed on the parameter card be used regardless of the results of each. A new method can start either with the final results obtained by the previous method or with the same initial estimates for each method. There is no return from MAXLIK between methods. The results of each method are printed in MAXLIK; only the last results obtained in the analysis are saved in core and returned in X and V, and optionally stored on disk.

Any number of analyses can be done in one run. Control returns to the user's calling program after each analysis. A new analysis is initiated by another call to MAXLIK in the calling program and another set of program parameter cards in the data deck. If the model is the same, the call to MAXLIK is the same; i.e., it may be included in a loop. If different models are to be analyzed in a run, subroutines corresponding to LOGLIK and DEPAR must be present for each model. All these subroutines must have unique names and the names must all be included in an EXTERNAL statement.

If the same data are to be used in different analyses they need not be read in again, provided the user does not destroy them in his log likelihood subroutine. If the same data are to be analyzed using different models, they must be present in the likelihood subroutine for each model; storing them in a COMMON region avoids the necessity of reading them in for each model.

C. User supplied subroutines

The subroutines LOGLIK and DEPAR are called by MAXLIK. Their structure and function are described below.

1. General description of LOGLIK

LOGLIK is the main subroutine for the model. It can have from one to seven sections:

(i) Initialization

(ii) Computation of the log likelihood corresponding to a given set of parameter values

(iii)-(vii) Optional computations and output.

Section (ii) is always necessary. The other sections are optional, allowing greater user interaction with MAXLIK. In particular, when an analysis is complete MAXLIK can call LOGLIK before and during the printing of the final results, in order to compute and print special extra results along with the results of the iteration. These sections are described in detail beginning p. IV-7 .

The subroutine statement is

```
SUBROUTINE LOGLIK(DL,X)
```

Program variables

DL Real (double precision) scalar. Resultant value of the log likelihood.

X Real (double precision) array. This array contains the values of the parameters for which the log likelihood is to be computed. These values are passed to LOGLIK by MAXLIK. The initial values can be computed in section (i) of LOGLIK described below, but otherwise only values of parameters which are functionally dependent may be changed in the subroutine.

In many problems variables in addition to those in the subroutine argument list must be passed between LOGLIK and MAXLIK; this is accomplished by including the following labeled COMMON statements in LOGLIK, and also, if necessary, in the driver program:

```
      COMMON/MLECAM/
     * XL(n),XU(n),FOPT(1Ø),FEST(n),Z(n),XTK,XTK2,
     * IFR(n),LRX(n),IOPT(4Ø),IDIST(2Ø),LABEL(n,7),KTRAN,NT,NR,NS
```

The dimension n in this statement must be identical to the dimension compiled in MAXLIK; see p. V- 1 for a discussion of this dimension. All real variables in this statement must be double precision if the double precision version of MAXLIK is used. Since the variables have a variety of different functions they will be described individually under the sections of LOGLIK in which they can be used.

* The * is punched in column 6 to indicate a continuation card.

Any variables not mentioned should be ignored by the user.

2. Sections of LOGLIK

The variable IOPT(14) which is included in the COMMON statement given above is set to a value from one to seven in MAXLIK, before each call to LOGLIK, to indicate which of the seven sections of LOGLIK is to be executed. If only section (ii) of LOGLIK is needed, i.e., the same computations are to be performed every time LOGLIK is called, this variable can be ignored.

The different sections of LOGLIK, the circumstances under which they are used, and relevant program variables are now described.

(i) Initialization

LOGLIK is called with IOPT(14)=1 once at the beginning of each analysis so that LOGLIK can include input, initialization of program variables, and computations,necessary for the evaluation of the log likelihood,that are independent of the values of the estimates. The log likelihood of the initial estimates must be computed and stored in DL during this first execution of LOGLIK; i.e., both sections (i) and (ii) must be executed when IOPT(14)=1. However, if no initialization is necessary, then no distinction is needed in LOGLIK between the call with IOPT(14)=1 and the call with IOPT(14)=2.

The following variables, all of which except X are included in the COMMON statement on p. IV-6 , can be computed, modified, or used in this section.

X Real array. Initial estimates for the t parameters.

XL Real array. Lower bounds for the t parameters.

XU Real array. Upper bounds for the t parameters.

On entry into section (i) of LOGLIK these three arrays will contain one of the following:

a. The values read in, if any, as program parameters, on cards 6, 7, and 8 in MAXLIK or read in using the option for unformatted input. (See p. IV-25, IV-35.)

b. The values used in the preceding analysis, if not destroyed in the driver program, or values computed in the driver program before the call to MAXLIK.

The following two arrays are printed with the summary output. Blanks will be inserted in any locations not filled by the user.

IDIST Integer array. This array can be filled with an alphanumeric title for the model. The title can fill up to 20 computer words.

LABEL 2-dimensional integer array. This array may be filled with alphanumeric labels for the individual parameters; up to seven computer words are allowed for each parameter.

LABEL(1,1)-LABEL(1,7) should contain the label for parameter 1,
LABEL(2,1)-LABEL(2,7) should contain the label for parameter 2, etc.

The following variables are read in as program parameters in MAXLIK and can be used but not changed in LOGLIK. Only the names of the FORTRAN variables are given here; for details see card 2, p. IV-16. These variables are all scalars.

Variable name and type	Contents	Columns
XTK Real (double precision)	Transformation constant	61-65
XTK2 Real "	Transformation constant	66-70
IOPT(1) Integer	KRDAT	58
IOPT(2) "	KREST	56
IOPT(3) "	KRBD	57
IOPT(4) "	NS	43-45
IOPT(5) "	NR	40-42
IOPT(6) "	NT	37-39
IOPT(7) "	KTRAN	59
IOPT(8) "	NFIX=NT-NR	
IOPT(9) "	MM=current iteration method	72, 73, or 74

The following two arrays are computed in MAXLIK from the values on program parameter input cards 4 and 5. They may be used but should not be changed in LOGLIK.

IFR Integer array. An array used to identify parameters involved in functional relationships:

IFR(i) = -2 if parameter i is fixed

IFR(i) = -1 if parameter i is functionally dependent

IFR(i) = 0 if parameter i is not involved in any functional relationships.

IFR(i) = 1 if parameter i is an independent parameter in one or more functional relationships.

LRX Integer array. An array indicating the status of parameters during the iteration:

LRX(i) = 0 if parameter i is independent and not fixed; normal iteration is performed on this parameter.

LRX(i) = 1 if parameter i is functionally dependent or fixed at the beginning of the analysis; no iteration is performed on this parameter.

LRX(i) = 2 if parameter i becomes temporarily fixed during the iteration.

(ii) Computation of the log likelihood

LOGLIK is called with IOPT(14)=2 for each evaluation of the log likelihood after the first in the analysis. The values of the functionally independent parameters are stored in X by MAXLIK. The values of functionally dependent parameters, however, are not computed automatically by MAXLIK, and so they must be computed each time sections (i) and (ii) of LOGLIK are called. This can be done by calling the subroutine DEPAR from LOGLIK.

MAXLIK checks automatically that all functionally independent parameters are within their bounds before calling LOGLIK. It may happen, however, that the log likelihood is not defined for a given set of values for the parameters, even though the individual parameters are all within bounds. If such a condition is detected in LOGLIK, the statements

```
IOPT(11)=1
RETURN
```

will cause MAXLIK to take the same action as it would if a parameter were outside the bounds, and then continue iterating. If this situation can occur, the COMMON statement on p. IV-6 must be included in LOGLIK.

Use of IOPT(11) also provides a limited method for handling more complex constraints on the parameters. Given that the initial estimates satisfy any set of constraints, each new set of values can be tested in LOGLIK, and, if the constraints are no longer satisfied, IOPT(11) can be set equal to 1 so that MAXLIK will try values closer to the last values which did satisfy them. A disadvantage of doing this is that, since the parameters are changed one at a time, the step sizes may get so small that some parameters become fixed, causing the test for convergence to be passed before arriving at a maximum.

In many calls to LOGLIK only one parameter has a value different from its value during the preceding likelihood computation, and in many other calls only a few of the parameters have different values. Thus in many cases considerable machine time can be saved by not re-computing parts of the log likelihood which are the same as the preceding time LOGLIK was entered. The user may want to save the values of the parameters each time section (i) or section (ii) is called in order to determine, the next time, which computations can be skipped.

The number of times the log likelihood is computed during the use of each method in an analysis can be counted by adding 1 to the variable IOPT(36) each time section (ii) of LOGLIK is executed. (IOPT(36) is in the COMMON statement, p. IV-6 .)

(iii) Transformation of final estimates

LOGLIK is called with IOPT(14)=3 if KTRAN>1 (program parameter card 2, column 59). If the estimates of the parameters were obtained using transformed data, LOGLIK can contain coding here to compute the values of the final estimates in the original scale. The final estimates computed during the iteration will be in the array X when LOGLIK is entered. These must be left unchanged, and the values in the original scale must be stored in the array Z which is in the COMMON statement described above, p. IV- 6. Both sets of estimates will then be printed automatically by MAXLIK. The transformation constants KTRAN, XTK, and XTK2 are also in the COMMON statement. See program parameter card 2, columns 59-70, p. IV-16, for an explanation of these variables.

(iv) Printing of data

LOGLIK is called with IOPT(14)=4 if KPRDAT>0. (parameter card 2, column 49 p. IV-16). If the user wishes to have his data printed along with the results, the coding to do the printing must be included in this part of LOGLIK. Any output coded in this section will be printed, starting on a separate page, immediately preceding the summary of results (see p. IV-31); and if multiple copies of the summary are specified, multiple copies of the data will also be printed.

(v)-(vi) Additional computations and output

LOGLIK is called with IOPT(14)=5 and with IOPT(14)=6 if KHYP>0 (parameter card 2, column 76, p. IV-16). These options allow additional computations such as tests of hypotheses, programmed by the user, to be performed and printed with the summary of results provided convergence was obtained. These calls are after the computation of the estimates and their variance-covariance matrix. Any computations should be done in (v) but the results of (v) are printed in (vi). The only advantages to including such computations here, rather than after the call to MAXLIK in the driver program, are that they will be performed after each method if more than one method is used in an analysis, and that the results will be printed on each copy if multiple copies are specified.

Various values computed in MAXLIK will be available in LOGLIK or in the driver program if the COMMON/MLECAM/ statement (see p. IV-6) and/or the additional COMMON statements described below are included. The following variables included in COMMON/MLECAM/ may be useful:

FOPT(3) log likelihood of the final estimates

FOPT(4) maximum last iteration change in estimates

FOPT(7) absolute value of the difference between the values of the log likelihood at the end of the last two iterations.

IOPT(10) exit code indicating the status of the iteration as follows:

1 The iteration has converged.

2 The iteration has not converged; another iteration will be performed.

3 The iteration has not converged but the maximum number of iterations specified by the user have been completed.

4 The iteration is converging to a minimum (possible in Newton-Raphson method only), or some error has occurred that makes continuation of the analysis impossible.

5 The direct search iteration converged but improved values were found applying the Newton-Raphson method to the final estimates from the search. This code is possible only after the direct search.

6\. Initial estimates unacceptable.

Note that section (v) of LOGLIK is called only if IOPT(10)=1. The other exit codes can be useful when additional computations are included in the driver program or if there is more than one analysis in a run.

IOPT(31) exit code indicating the result of the computation of the variance-covariance matrix as follows:

0	The variance-covariance matrix was computed successfully.
1	Round-off error occurred in the matrix of numerical second partial derivatives, but the variance-covariance matrix was computed.
2	Negative elements occur on the diagonal of the variance-covariance matrix.
3	The matrix of second partial derivatives is singular.
4	The matrix of second partial derivatives could not be computed.
5	The variance-covariance matrix was not computed because convergence was not obtained.
6	The user specified that the variance-covariance matrix was not to be computed (program parameter card 2, column 53, p. IV-16.)

IOPT(12) number of parameters that converged to a bound. (If it is necessary to know which parameters converged to a bound, this information is available in the array LRX, which is also included in the COMMON statement. LRX(i)=2 at the end of the analysis if parameter i has converged to a bound.)

IOPT(15) Program parameter card 2, column 76 (p. IV-22) (KHYP).

The matrix of numerical second partial derivatives will be available in LOGLIK and/or the driver program if the following statement is included:

COMMON/DERIVM/D(v^2)

The dimension v^2 is discussed on p. V-1 . D is an m×m matrix stored in packed rectangular form, where m=r-s-(number of parameters that have converged to a bound).

The variance-covariance matrix is already available in V in the driver program. It will be available in LOGLIK if the following statement is included:

COMMON/VCVM/VC(v^2)

VC is an r×r matrix stored in packed rectangular form.

(vii) Additional labeling

LOGLIK is called with IOPT(14)=7 if KINF>0 (parameter card 2, column 75). This call occurs after the printing of the run identification and allows the user to print information in addition to the standard identification provided by printing the arrays ID and IDIST which are described on p. IV-15, IV-8.

This section can also include any computations which should be performed at the end of each method used, regardless of whether convergence was obtained.

3. Description of DEPAR

Coding for the evaluation of any functionally dependent parameters must be included by the user in the subroutine DEPAR. If it is necessary to check whether the values of the dependent parameters are within their bounds, or satisfy other specified restrictions, such tests must also be included in DEPAR. The bounds are available in the COMMON/MLECAM/ statement described on p. IV-6, which can be included in DEPAR.

The subroutine statement is

SUBROUTINE DEPAR(X, LEXIT)

Program variables

X Real (double precision) array. Upon entry into DEPAR, X contains the current values of the independent parameters. The values of all dependent parameters must be replaced by the resultant new values computed in DEPAR. The independent parameters must not be changed in DEPAR.

LEXIT Integer scalar. Resultant exit sentinel to be set by the user as follows:

0 if the new values of all dependent parameters are within bounds and satisfy any specified restrictions

1 if the new value of any dependent parameter is outside the parameter's bounds or fails to satisfy any specified restrictions.

D. Program input

Input of the data to be analyzed must be programmed by the user either in LOGLIK or in the driver program.

The program parameters described below are read in by the control subroutine in MAXLIK using data set reference number 1. The parameters are normally on cards but could also be on disk or tape in card image format. There is also an option to read most of the parameters in unformatted; this is described on p. IV-35. A set of the parameter cards given below must be supplied for each call to MAXLIK.

All numbers read with I format are integers and must be right-justified in their field, with no decimal punched. All numbers on the parameter cards read with F format should have the decimal punched (or if integers they may be simply right-justified in the field). The decimal may be punched anywhere within the field and will override the given format. Variables read in A format may contain any valid alphanumeric codes.

Card 1 - Analysis Identification

Columns	Format	Variable Array	Description
1-80	20A4	ID	Arbitrary analysis identification which is printed at the beginning of the output and on the output summary. It may contain any alphanumeric punches. If columns 1-4 on this card are all blank, the run stops. Unless termination is desired, this card must have at least one character in one of the first four columns.

Card 2 - Iteration Parameters

Columns	Format	Variable Name and Range	Description
1-10	F10.0	EPS$\geq$0	Requested maximum relative change in the estimates between the final two iterations (ε). If EPS is 0 or blank on the card it is set = 0.001.
11-20	F10.0	EPS2	κ (p. III-5). If the maximum absolute difference between any of the log likelihoods calculated during an iteration and that found on the end of the previous iteration $\leq \kappa$, the iteration stops.
21-25	F5.0	D<1	Original step size δ. If D=0 or is blank on the card, it is set = 0.5. If D is negative, a value of δ will be read in for <u>each</u> parameter (card 3).
26			Blank (not used).
27	I1	4$\leq$KIN$\leq$9 or KIN=0	Data set reference number if unformatted parameters are to be read. These unformatted parameters must have been written by a previous run of this program. (See p. IV-35.) If KIN=0, no unformatted parameters are read.
28-31	I4	0$\leq$IDI$\leq$9999	Identification number for the particular set of unformatted parameters to be used. The program will locate the set of parameters with this number written as output identification by a previous run. (See columns 77-80.) IDI is ignored if KIN=0.
32			Blank (not used).

33	I1	$4 \leq KOUT \leq 9$ or $KOUT=0$	Data set reference number for binary output of program parameters and results. If KOUT=0 there is no unformatted output of parameters and results.
34-36	I3	$NIT \geq 0$	Maximum number of iterations to be done in each method tried.
37-39	I3	$1 \leq NT \leq NTMAX$ and $NT \leq m$	Total number of parameters which may be estimated in the model (t). NTMAX contains the maximum t allowed by the package (see p. V-1, VI-1) and m contains the maximum t allowed by the driver program (see p. V-1, VI-1 and IV-4).
40-42	I3	$1 \leq NR \leq NT$ and if col. 53=1, $NR \leq NTVMAX$ and $\leq u$	Number of parameters to be estimated in this analysis (r). This is the number <u>not</u> held fixed at a constant value. See p. V-1, VI-1 and IV-4.
43-45	I3	$0 \leq NS < NR$	Number of functionally dependent parameters specified for this analysis (s).
46-47	I2	$NCOPY \geq 1$	Number of copies of output summary pages to be printed. Only one copy of the rest of the output is printed.
48	I1	KPR=0 or 1	If KPR=1, the results of each iteration will be printed. (See section 2, p. IV-30). Otherwise they will be omitted.
49	I1	KPRDAT=0 or 1	If KPRDAT=1, the data are printed, preceding the summary output, by part (iv) of LOGLIK (see p. IV-31).
50			Blank (not used).
51	I1	MEST=0 or 1	After each method of iteration used, either the original estimates or the final estimates are saved so that they can be used as initial estimates for another method.

			If MEST=0, the original estimates are saved. If MEST=1, the final estimates will be saved. If another method is used in this analysis, whichever set of estimates is saved will be used as initial estimates for the new method. If only one method is used either value is satisfactory. The best values obtained by any method are saved at the end of the analysis for return to the driver program, regardless of the value of MEST. The initial estimates are not saved at the end of the analysis.
52	I1	MUSE=0 or 1	If MUSE=0, another method is tried only if the last method failed to give convergence. If MUSE=1, all methods listed in the array METHOD (columns 73-75) are tried regardless of the results of each. Either value is satisfactory if there is only 1 method.
53	I1	MDERIV=0, 1, 2 or 3	If MDERIV=0, the variance-covariance matrix of the estimates is not computed. If MDERIV=1, the variance-covariance matrix is computed if convergence is obtained by direct search or as needed if the Newton-Raphson method is used. If MDERIV=2, the variance-covariance matrix is recomputed at the end of method 2 (see p. IV-21) if improved results are obtained. If MDERIV=3, the variance-covariance matrix is recomputed at the end of method 2 if improved results are obtained and more than one iteration is required.

54-55 Blank (not used)

56 I1 KREST=0 or 1

If KREST=0, initial estimates are not read in by MAXLIK (card 6, p. IV-25). In this case they must be supplied in one of the following ways:

a. The final estimates from the preceding analysis can be used if the contents of the array X (see p. IV-6) are not destroyed in the driver program between calls to MAXLIK.

b. They can be read in or computed in the driver program.

c. They can be read by using the option to read the estimates written during a previous run (p. IV-35).

d. They can be read or computed in section (i) of LOGLIK.

If KREST=1, initial estimates are read in by MAXLIK (card 6). Note that when different methods are specified in one analysis, the initial estimates for methods other than the first are always the initial or final estimates from the preceding method. No new initial estimates can be read or computed without starting a new analysis.

Initial estimates read in on card 6 will override initial estimates supplied in a, b, and c above but not those supplied in d.

57 I1 KRBD=0 or 1

If KRBD=0, lower and upper bounds for the estimates are not read in by MAXLIK for this analysis. In this case they must be supplied in one of the ways given above for initial estimates.

If KRBD=1, both lower and upper bounds are read

			in by MAXLIK (cards 7 and 8, p. IV-25).
58	I1	$0 \le KRDAT \le 9$	This variable has no effect on MAXLIK but is included for the user's convenience. It is stored in IOPT(1) (see p. IV-6) so that it is available in LOGLIK and DEPAR provided the COMMON/MLECAM/statement is included in these subroutines. It can be ignored or used in any way convenient for the user. For example, it could be used to control input of data to be analyzed, which is supplied entirely by the user, by coding it 0 if new data is to be input for this analysis 1 if the data from the preceding analysis is to be used.
59	I1	$0 \le KTRAN \le 9$	Transformation code. If KTRAN>1, the maximum likelihood package calls section (iii) of LOGLIK to obtain values of the estimates in the original scale (see p. IV-10). It has no other effect on MAXLIK, but is included in the COMMON/MLECAM/statement so that it is available in LOGLIK to indicate that the data are to be transformed. The actual transformation must be programmed by the user. If the data are transformed but the values of the estimates are not changed by this transformation, section (iii) of LOGLIK can consist merely of storing the values of the array X in the array Z.
60			Blank (not used)

61-65	F5.0	XTK	These variables have no effect on MAXLIK
66-70	F5.0	XTK2	but are for the convenience of the user. They are available in LOGLIK and DEPAR provided the COMMON/MLECAM/ is included in these subroutines. They can be used, for example, for constants involved in data transformations.
71	I1	$1 \leq NM \leq 3$	The maximum number of methods to be used sequentially to obtain the maximum likelihood estimates in this analysis.
72	I1	$0 \leq METHOD(1) \leq 6$	Code number of the first method to be used.
73	I1	$0 \leq METHOD(2) \leq 6$	Code number of the second method to be used, if the number of methods (column 71)$\geq$2.
74	I1	$0 \leq METHOD(3) \leq 6$	Code number of the third method to be used; if the number of methods (column 71)=3.

The various methods and their corresponding code numbers are:

0 No iteration. Compute likelihood of initial estimates only.

1 Newton-Raphson method recomputing the numerical second derivatives in each iteration.

2 Newton-Raphson method without recomputing the second derivatives.

3 Direct search of the likelihood surface, including the 2^{r-s}-dimensional search around the estimates at convergence. If convergence is obtained and the variance-covariance matrix is computed (see column 53), one iteration of the Newton-Raphson method is performed; if this improves the estimates significantly method 2 is automatically used to try to improve them further.

4 This method is the same as method 3 except that the 2^{r-s}-dimensional search around the final estimates is omitted. This method is recommended for models involving a large number of parameters or lengthy log likelihood computation.

5 Start with the 2^{r-s} dimensional search around the estimates and then continue with method 3. This might be useful if method 4 was specified in a previous run and then the search was wanted at a later date.

6 Compute the variance-covariance matrix after reading in the estimates obtained in a previous run. One iteration of the Newton-Raphson method is performed automatically. If this improves the estimates significantly, method 2 is then used provided the number of iterations specified (NIT, columns 34-36) is greater than 1.

The methods specified in columns 72-74 are used sequentially. If MUSE (column 52) = 0, a new method is used only if the preceding one did not converge. If MUSE=1 all methods listed will be used.

75	I1	$0 \le KINF \le 9$	If KINF>0, the maximum likelihood package calls section (vii) of LOGLIK to print additional run identification. KINF is stored in IOPT(37) (see p. IV-14) so that it is available in LOGLIK and DEPAR provided the COMMON/MLECAM/ statement is included in these subroutines. (see p. IV-6).
76	I1	$0 \le KHYP \le 9$	If KHYP>0, MAXLIK calls sections (v) and (vi) of LOGLIK so that additional computations, such as tests of hypotheses, can be done. KHYP is stored in IOPT(15) (p. IV-11) so that is is available in LOGLIK and DEPAR provided the COMMON/MLECAM/ statement is included in these subroutines. (see p. IV-6).
77-80	I4	$0 \le IDOUT \le 9999$	Identification number if results of this run are to be written in an unformatted data set. This number identifies the results of a particular analysis for future retrieval (see IDI, columns 28-31); therefore it should be unique for each analysis in a given run and also

unique for analyses in different runs whose results are written in the same data set.

Card(s) 3 - Initial Steps Sizes for Each Parameter (Optional)

This card is included only if D (card 2, columns 21-25) is negative. In this case the numerical value of D is ignored and instead a vector of values δ_i, i=1, 2, ..., NT, is read in. This option allows continuation of an analysis exactly where it stopped in a previous run by using the final step sizes printed as initial step sizes for the new run.

Column	Format	Variable Name and Range	Description
1-10	F10.0	0<DEL(1)<1	Initial step size for parameter 1.
11-20	F10.0	0<DEL(2)<1	Initial step size for parameter 2.
⋮			
71-80	F10.0	0<DEL(8)<1	Initial step size for parameter 8.

Continue on as many cards as necessary, same format.

Card(s) 4 - Parameters Fixed at a Constant Value (Optional)

If NT = NR on card 2, no parameters are fixed and this card must be omitted.

If NT > NR, this card must be included and must contain NT - NR numbers.

Columns	Format	Variable Name	Description
1-3	I3	LFIX(1)	First parameter fixed in this analysis
4-6	I3	LFIX(2)	Second parameter fixed
⋮			
73-75	I3	LFIX(25)	25th parameter fixed

Parameters are numbered from 1 to NT and may appear in this list in any order. Continue on as many cards as necessary, same format.

Parameters are fixed at their initial estimates.

Card(s) 5 - Functional Relationships (Optional)

These cards must be included if NS (card 2, columns 43-45) > 0. One of these cards must be present for each functionally dependent parameter; thus there will be NS cards.

Columns	Format	Variable Name and Range	Description
1-3	I3	$1 \le IFDP \le NT$	Parameter number of the functionally dependent parameter.
4-6	I3	$1 \le NFIDP \le NT-1$	Number of functionally independent parameters involved in this functional relation
7-9	I3	$1 \le LFIDP(1) \le NT$	Number of the first independent parameter used to define the dependent parameter IFDP
10-12	I3	$1 \le LFIDP(2) \le NT$	Number of the second independent parameter used to define the dependent parameter IFDP
⋮	I3	⋮	
73-75	I3	$1 \le LFIDP(23) \le NT$	Number of 23rd independent parameter

NFIDP independent parameters should be listed on this card. If NFIDP<23, the remainder of the card should be blank. If NFIDP>23, the list should continue with the I3 format in column 1-3 of the next card.

Any number of the parameters may be either defined by functional relationships or fixed. However a parameter cannot be declared to be functionally dependent and fixed at the same time. If all of the independent parameters in a functional relationship are fixed, then the dependent parameter should be declared fixed rather than dependent, and the number of functionally dependent parameters NS reduced. Parameters that can be independent parameters in a functional relationship, but that are fixed in a particular run, should not be included in the list LFIDP for that run.

Also, only independent parameters should be listed as arguments of each dependent parameter, even though it may actually be computed using a previously defined dependent parameter.

Card(s) 6 - Initial Estimates (Optional)

These cards must be present if KREST=1 (card 2, column 56); otherwise they must be omitted.

Columns	Format	Variable Name	Description
1-10	F10.0	EST(1)	Initial estimate for parameter 1
11-20	F10.0	EST(2)	Initial estimate for parameter 2
⋮			⋮
71-80	F10.0	EST(8)	Initial estimate for parameter 8

Continue on as many cards as necessary.

Cards 7 and 8 - Lower and upper bounds for the estimates (Optional)

If these cards are present, KRBD (card 2, column 57) must be 1.

If they are not present, the bounds from the preceding analysis are used unless the user overrides these by storing new lower and upper bounds in the arrays XL and XU respectively by one of the methods given on p. IV-19.

Columns	Format	Variable Name	Description
1-10	F10.0	XL(1)	Lower bound for parameter 1
11-20	F10.0	XL(2)	Lower bound for parameter 2
⋮			⋮
71-80	F10.0	XL(8)	Lower bound for parameter 8

Continue the lower bounds on as many cards as necessary.

The lower bounds are followed by the upper bounds which are read into the array XU. Their format is identical to that of the lower bounds.

Both lower and upper bounds must always be present for each parameter. No values outside of the bounds will be used, but the values that are used will not necessarily cover the entire range between the bounds; therefore if a parameter does not actually need bounds, they can be made arbitrarily wide.

It is preferable to avoid setting a bound equal to zero, since this sometimes leads to an excessive amount of computation. Setting the bound equal to $1+\varepsilon/2$ or $1-\varepsilon/2$, for example, is often a reasonable alternative.

Card(s) 9 - Data to be analyzed

Cards 1-8 above are followed by any data the user reads in section (i) of LOGLIK using data set reference number 1. MAXLIK has no provisions for reading in the actual data to be analyzed or for transforming it. The data must either be read in section (i) of LOGLIK or be included in a COMMON area in LOGLIK other than MLECAM. The four variables KRDAT, KTRAN, XTK, and XTK2 on card 2 described above, are included for convenience in handling data input and transformation, but do not cause data to be read or transformed in MAXLIK. These variables are available in LOGLIK if the COMMON/MLECAM/ statement is included.

Note that it is advisable to store all the data in core, if at all possible, so that they do not have to be read each time the log likelihood is computed.

The cards 1-9 may be repeated for as many analyses as desired. There must be a call to MAXLIK corresponding to each analysis. Cards 1 and 2 must be present for each analysis; the others are optional, depending on the parameters on card 2.

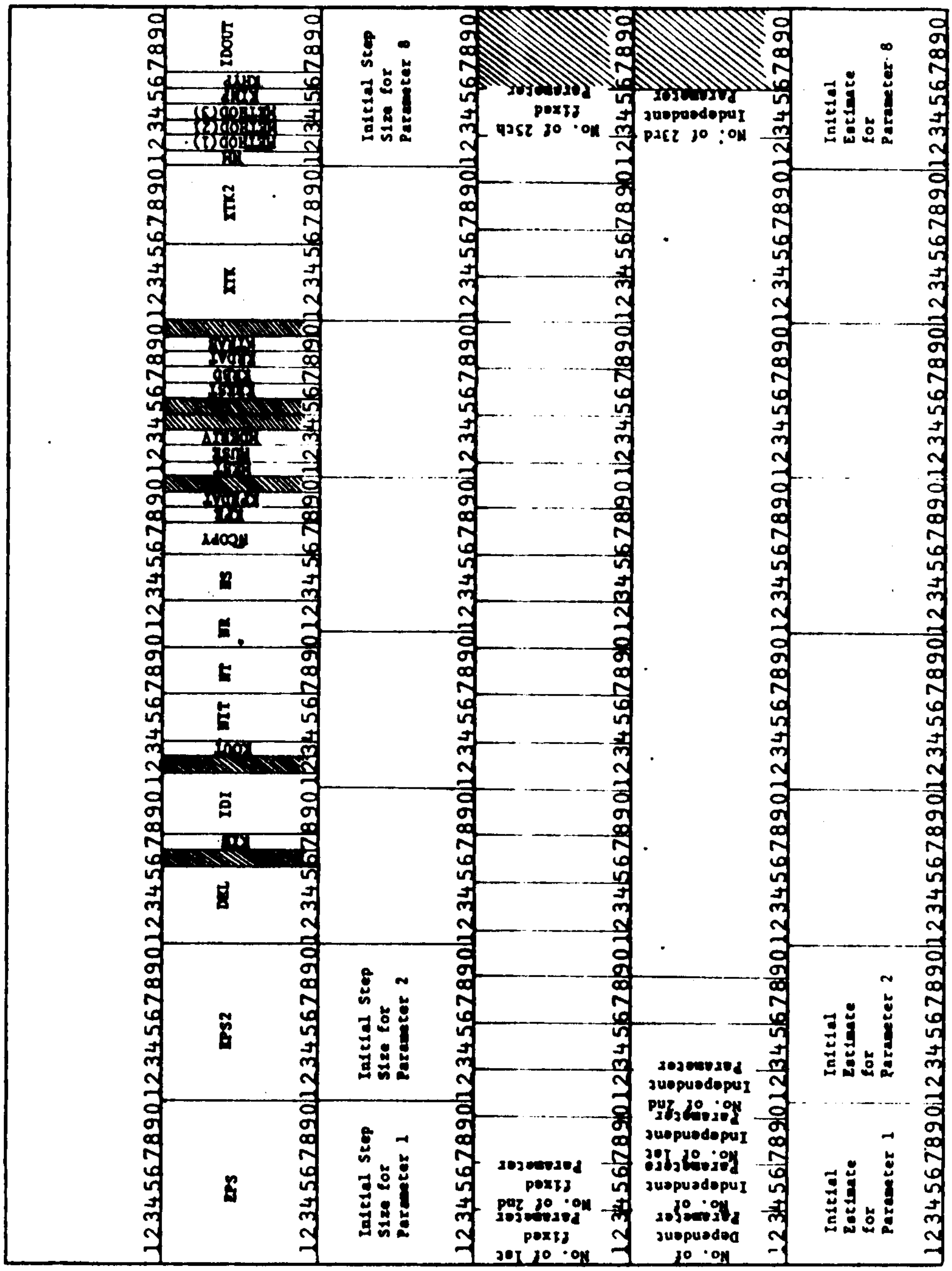
Card 10 - End of run indication

Blanks in columns 1-4 of card 1 (the identification card) for an analysis indicate end of run. Alternately, the user can test for end of run in the driver program.

Summary: Card Input

The program input is summarized on this and the following page.

MULTIPLE CARD INPUT



1. Analysis Identification
   Always present

2. Iteration Parameters
   Always present

3. Initial Step Sizes
   $[\frac{NT-1}{8} + 1]$ cards
   present if D<0

4. List of Fixed Parameters
   $[\frac{NT-NR-1}{25} + 1]$ cards
   present if NT-NR>0

5. *Functional Relation
   NS cards present
   if NS>0

6. Initial Estimates
   $[\frac{NT-1}{8} + 1]$ cards
   present if KREST=1

* actually NS sets - a varying no. of cards might be present for each relation

MULTIPLE CARD INPUT

7. Lower Bounds
$[\frac{NT-1}{8} + 1]$ cards
present if KRBD=1

Lower Bound for Parameter 1	Lower Bound for Parameter 2						Lower Bound for Parameter 8
1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

8. Upper Bounds
$[\frac{NT-1}{8} + 1]$ cards
present if KRBD=1

Upper Bound for Parameter 1	Upper Bound for Parameter 2						Upper Bound for Parameter 8
1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

9.

Insert any cards read in the user supplied subroutine LOGLIK here
12345678901234567890123456789012345678901234567890123456789012345678901234567890

Repeat 1. and 2., and if needed 3.-9. if there is another analysis
12345678901234567890123456789012345678901234567890123456789012345678901234567890

10. End of Run

Blank	
1234	567890123456789012345678901234567890123456789012345678901234567890123456789 0

12345678901234567890123456789012345678901234567890123456789012345678901234567890

E. Program output

All printed output is written using data set reference number 3. There are five types of printed output.

1. Initial Printout. This is especially useful if the run does not go to completion. It includes

a. Information read in on program parameter cards.

b. Initial estimates, bounds, the arrays IFR and LRX indicating the status of each parameter, and the log likelihood corresponding to the initial estimates.

c. The iteration method being used.

d. Messages indicating when a parameter is fixed at a bound, whether or not convergence is obtained, and the number of iterations necessary for convergence.

e. $-W^{-1}$. This is the variance-covariance matrix before rows and columns are inserted corresponding to parameters which are dependent or which have converged to a bound.

f. Check on the inverse: $W\ W^{-1}$.

g. Various error messages. See p. IV-34 for a description.

2. Results of each iteration. This output is optional (see KPR - card 2 - column 48, p. IV- 17). It will enable the user to continue a run exactly where it left off if for any reason it does not go to completion, and will also give an indication of how well the iteration method is working.

This output includes the iteration number, the old and new values of the estimates, the step size used for each parameter in the direct search of the log likelihood surface, the values of the first partial derivatives in the Newton-Raphson method, the old and new values of the likelihood and the maximum error.

3. Printout of the data. This is optional (see KPRDAT - card 2, column 49, p. IV- 17) and if used must be programmed as section (iv) of the subroutine LOGLIK.

4. Summary of results. These are the results generally wanted for a permanent record. They are printed for each method used and include:

a. Model identification if it is provided in the subroutine LOGLIK.

b. Any information printed in section (vii) of LOGLIK.

c. Run identification (card 1 of input).

d. Method used to compute the maximum likelihood estimates.

e. Lower bounds, initial estimates, and upper bounds for each parameter, labeled if parameter labels are supplied in the subroutine LOGLIK.

f. Initial step size(s) used.

g. List of parameters that are fixed.

h. List of parameters that are functionally dependent.

i. Final estimates of the parameters, labeled if labels are supplied in LOGLIK.

j. Standard errors of the estimates, provided the variance-covariance matrix has been comptued.

k. Log likelihood corresponding to the final estimates of the parameters.

l. Maximum last iteration change in the estimates.

m. List of parameters, if any, that have converged to a bound.

n. Number of iterations required for convergence, or the number performed if convergence is not obtained.

o. If the estimates are computed using the direct search algorithm, a message is printed indicating whether or not the results have been verified by the search around the final estimates.

p. If one iteration of the Newton-Raphson method has been applied to the final estimates from the direct search, the results of this iteration are printed. If the iteration results in significantly improved estimates, the program continues with method 2 (see p. IV-21) automatically, but the results from method 2 are printed separately.

q. Variance-covariance matrix of the estimates, if it has been computed.

r. If round-off error has been detected during the computation of the partial numerical derivatives, a warning is printed along with the final three approximations to the derivative.

s. Numerical first and second partial derivatives of the log likelihood function, if they have been computed.

t. Step sizes used in the last iteration of the direct search, if convergence has not been obtained.

5. Additional output. Any output programmed by the user in section (vi) of LOGLIK will be printed following the summary of the results.

Multiple copies of 3-5 can be printed (see NCOPY - card 2, columns 46-47, p. IV- 17).

In addition to the printed output, the user can specify unformatted output of the results (see p. IV-35).

F. Summary of user supplied information

The user of MAXLIK must provide statements to call it (from a main program or subroutine), the two subroutines LOGLIK and DEPAR, program parameter cards to be read by MAXLIK, and data cards to be read by LOGLIK, if necessary.

The specific functions performed by the user supplied portions of the package together with the corresponding locations are as follows:

Function		Location
1. Specify method of search of likelihood surface (including desired precision, step size etc.)		MAXLIK input card 2
2. Read data, transform if necessary, and perform any other preliminary computations desired.		Section (i) of LOGLIK
3. Provide initial estimates of parameters,		
either directly		MAXLIK input cards 2 and 6
or computed		Section (i) of LOGLIK
4. Provide algorithm for computing log likelihood of the transformed data, given a set of values of the parameters		Section (ii) of LOGLIK
5. Specify and test constraints on parameters		
a) Upper and lower bounds:	specify:	MAXLIK input cards 2,7,8 or LOGLIK, section (i)
	test:	DEPAR
b) Parameters held fixed		MAXLIK card 4
c) Functional relations among parameters:	specify:	MAXLIK cards 2 and 5
	compute:	DEPAR
d) Other constraints		DEPAR; LOGLIK section (ii)

Function	Location
6. Final computations following completion of iteration	
a) Transform final estimates	MAXLIK card 2, and LOGLIK, section (iii).
b) Any additional computations with estimates	MAXLIK card 2, and LOGLIK, section (v).
7. Output	
a) Analysis identification	MAXLIK card 1
b) Model identification	LOGLIK, sections (i) and (vii)
c) Data	LOGLIK, section (iv)
d) Parameter labels	LOGLIK, section (i)
e) Number of summary pages, etc.	MAXLIK card 2
f) Results of additional computations performed in LOGLIK	LOGLIK, section (vi)
8. Indication that run is to stop	Driver program, or MAXLIK card 1

G. Error indications

The list below includes errors which are detected by the program and the resulting action taken; in each case a message is printed.

1. Program dimensions too small. If there are more parameters in the model than the program allows, the program stops without returning from MAXLIK to the driver program. No recovery is possible. To remedy this situation see p. VI-1.

2. Initial estimates are unacceptable. In this case either the initial estimates are not within bounds or some user specified restriction is violated by the initial estimates, as indicated by IOPT(11)=1 after return from the first call to the subroutine LOGLIK. When this occurs, the exit code IOPT(10) is set

equal to 6 and control is returned to the driver program. At this point all data for this analysis have been read so the program can continue with another analysis in the same run.

3. Repeated entry into search around final estimates. Since the 2^{r-s}-dimensional search can be very time consuming, it will be done no more than twice during each direct search. If improved values are found the second time, a warning is printed but the analysis proceeds normally.

4. Matrix of second derivatives could not be computed. The analysis proceeds normally, omitting computation of the variance-covariance matrix. IOPT(31) is set equal to 4.

5. Variance-covariance matrix is singular. The analysis proceeds normally, omitting computation of the variance-covariance matrix. IOPT(31) is set equal to 3.

6. Round-off error in variance-covariance matrix. This is printed as a warning only. The user should compare the three values of the numerical derivatives printed to see whether the discrepancies are large enough to invalidate the variance-covariance matrix; IOPT(31) is set equal to 1.

7. Negative variances are printed in the standard error column. See p. III-12 for discussion. IOPT(31) is set equal to 2, and the analysis procedes normally.

8. Likelihood subroutine is not reusable. This means that the likelihood subroutine does not give identical results when called with identical values of the estimates of the independent parameters, first with IOPT(14)=1 and then with IOPT(14)=2. (See p. IV-7.) This can happen if:

   a. The subroutine is not properly initialized each time section (ii) is executed.

b. There are functional relationships specified, but the dependent parameters are not computed each time section (ii) is executed.

If this error occurs, the run is terminated without returning from MAXLIK to the driver program.

Even though a log likelihood subroutine works correctly in test runs, it may be inaccurate for particular combinations of data and parameter estimates. The user should include coding to avoid such errors as division by zero, taking the log of zero or a negative number, and other situations that cause the likelihood to be undefined. See p. IV-9 for directions on how to handle this situation. If underflow or overflow errors occur in the log likelihood subroutine, the user must decide whether or not their effect on the log likelihood is sufficient to affect the accuracy of the results.

H. Unformatted Input and Output

If KOUT (card 2, column 33) $\geq$ 4, selected input information and results will, at the end of the analysis, be written in the data set with reference number KOUT; this is written without format control, and is in addition to the printed output. It can be used for continuing an analysis at a later time or as input to other programs. A message that this output has occurred is printed as follows:

RESULTS WRITTEN ON DATA SET NO. _____.

ID NUMBER _____.

The ID number printed is the number IDOUT (card 2, columns 77-78); it is stored in IOPT (33) and included in the data set for use in retrieving the results from this particular analysis.

The following arrays are written:

Array	Contents	Page References
IOPT	Miscellaneous information	IV-8, IV-12, IV-13
ID	Run identification	IV-15
IDIST	Model identification	IV-8
X	<u>Final</u> estimates	IV-6
FOPT	Miscellaneous information including especially the final log likelihood in FOPT (3)	IV-12
DEL	<u>Final</u> step sizes	IV-23
XL, XU	Lower bounds, Upper bounds	IV-7
LRX	List used to determine whether a parameter is independent, functionally dependent, or fixed	IV-9
IFR	List used to identity parameters involved in functional relationships	IV-9
STDE	Standard errors (zeroes if not computed, variances if negative)	III-10
V Provided IOPT(31)$\leq$2	Variance-covariance matrix	IV-12 - IV-14
D Provided IOPT(31)$\leq$3	Negative second derivatives	IV-12 - IV-14

If the numerical second derivatives and the variance-covariance matrix have been computed but are not wanted in this data set, they can be suppressed as follows:

1. Set IOPT(31) = IOPT(31) + 7 in part (vii) of LOGLIK;

and 2. Code KINF (card 2, column 75) $\geq$ 1, so that part (vii) will be executed.

Results of different analyses can be written in the same data set provided IDOUT is unique for each analysis.

If KIN (card 2, column 27) $\geq$ 4, the program will search the data set with reference number KIN for the set of records (written during a previous run) with identification number IDI (card 2, columns 28-31), and will read in this set of records. If KIN = 9, the data set is rewound afterwards; otherwise it is not. This input follows the reading of cards 1 and 2, which must still be present for each analysis. A message that this input has occurred is printed as follows:

INPUT FROM DATA SET NO. ______

ID NUMBER ______.

Both the new run identification (card 1) and the run identification stored in the data set from the previous run are printed with the results.

The information read in from the earlier analysis is used as follows:

1. If D (card 2, columns 21-25) $\geq$ 0, the final step sizes from the earlier analysis are used as initial step sizes. If D < 0, new step sizes are read in for each parameter (card(s) 3).
2. If KREST (card 2, column 56) = 0, the final estimates from the earlier analysis are used as initial estimates, unless the user overrides them in part (i) of LOGLIK. If KREST = 1, new initial estimates are read in (card(s) 6).
3. If KRBD (card 2, column 57) = 0 the bounds from the earlier analysis are used unless the user overrides them in part (i) of LOGLIK. If KRBD = 1, new bounds are read in (cards 7 and 8).
4. If the first method to be used in the analysis is the Newton-Raphson method (method 1 or 2, see p. IV-21), the variance-covariance matrix is computed afresh.

If a set of records with IDI as identification is not found in the data set, an error message is printed and the run is terminated. Thus if KIN $\neq$ 9, a set of records will not be found if it precedes the last called set.

The subroutine RDUNF which reads the unformatted data in MAXLIK can be used to read these data in other FORTRAN programs. The use of this subroutine is described by comments on the program listing.

V. RESTRICTIONS AND RUN CONSIDERATIONS

A. Number of parameters

The number of parameters which can be estimated for a model is limited by computer time and storage considerations, as well as by numerical difficulties associated with estimating a large number of parameters. On the IBM 360/75 and 370/165 the program has been used successfully to estimate up to 32 parameters. The maximum number of parameters allowed by the package is determined by the dimension statements as described on p. VI-1.

B. Program size

MAXLIK requires approximately 21,000 words of storage for the program statements, including the necessary system subroutines, scalar variables, arrays not dependent on the number of parameters, and constants. (This figure may be different at different installations.) The additional storage required for arrays is dependent on the maximum number of parameters allowed by the program. There are two Fortran variables, NTMAX and NTVMAX, which are set equal to constants in the subroutine MAXLIK, at the time it is compiled, as follows:

NTMAX = n = the maximum total number of parameters allowed in a model.

NTVMAX = v = the maximum number of parameters that can be estimated in a model when the variance-covariance matrix is to be computed.

The amount of storage required for arrays is, in the current version,

$2(10v^2 + 23n)$ words for real double precision arrays plus

$v^2 + 15n$ words for integer arrays.

If single precision is used, the number 2 in the first formula is omitted.

See section VI for ways of decreasing the amount of storage necessary.

C. Estimation of run time

Most of the time is normally spent evaluating the log likelihood. Therefore, although it is impossible to give any general time estimate, a rough estimate for a given model and a given set of data can be obtained by determining the time needed to evaluate the log likelihood once, and then using the following considerations to determine the probable number of evaluations.

1. During each basic iteration of the direct search of the log likelihood surface there are usually 2 or 3 evaluations per parameter being estimated. There may however be more, as when the formula in *e*, p. III- 4 , does not give improved results, when a parameter converges to a bound, or when the values of the log likelihood are not defined for some parameter points.

2. During the supplementary iteration after each basic iteration there are j+1 additional likelihood computations where

$$2^j\underline{\theta} - (2^j-1)\underline{\theta}_o = \underline{\theta}^*_j$$

gives the best values of the parameters in this step.

3. 2^{r-s} evaluations of the likelihood are performed each time the search around the final estimates is done. Therefore this step, while extremely useful for ensuring that the estimates are actually near maximum, is impractical for models having a large number of parameters and/or lengthy likelihood computations.

4. The computation of the variance-covariance matrix and the first calculation of the first partial derivatives usually involves at least $6(r-s)^2$ evaluations of the log-likelihood -- but there may be fewer evaluations if any parameters have converged to a bound.

5. Using the Newton-Raphson method at the end of the search requires only one extra likelihood evaluation for the first iteration. If further iterations

are done, each requires at least 6(r-s) evaluations; and more are required if the variance-covariance matrix is recomputed at the end (see 4.).

6. For the Newton-Raphson method recomputing the variance-covariance matrix in each iteration, the number of likelihood evaluations per iteration is the same as the number required for the computation of the variance-covariance matrix, plus one.

D. Accuracy of results

The accuracy of the results depends on the accuracy with which the likelihood is evaluated. Double precision is strongly recommended whenever practical; it is especially important for the computation of the variance-covariance matrix.

Even if an analysis is completed successfully, there is no guarantee that

a. the parameters are actually at a local maximum, and

b. if they are at a local maximum, that this is the absolute maximum.

If the results from the direct search have been verified by the 2^{r-s} dimensional search around the final estimates, it is very unlikely that the parameters are not near a local maximum. However, when the number of parameters is large, these methods of verification can be prohibitively time consuming. The following procedures may help verify the results more efficiently and may also help locate an absolute maximum, although there is never any way of ensuring that the parameters are at an absolute maximum.

1. Repeat the analysis using different initial estimates.
2. Repeat the analysis using a different iteration method.
3. Apply a different iteration method to the final estimates obtained by the first iteration method.
4. Apply the same method using the final estimates as initial estimates.

See pp. III-12 and IV-33 for a discussion of specific problems that may occur.

VI. POSSIBLE PROGRAM CHANGES

A. Dimension Changes

The program dimensions can be increased to handle larger models or decreased to conserve core storage. To do this, change the subroutine MAXLIK (the first subroutine in the package) as follows:

To change n, as defined on p. V-1, change the value of the Fortran variable NTMAX and change the dimensions of the arrays whose dimensions are n or n+1. The dimensions that are n in the COMMON statements throughout the program must also be changed.

To change v, as defined on p. V-1, change the value of the Fortran variable NTVMAX and change those dimensions that are v^2 or $3v^2$.

If the dimensions are changed in MAXLIK, the dimensions used in the COMMON statements (see pp. IV-6, IV-13, IV-14), if these are included in the driver program or in the user supplied subroutines, must be changed accordingly. Note that n and v give maximum values for a compiled version of the package; the actual number of parameters in a given run are read in on parameter card 2, (p. IV-16).

B. Omitting the variance-covariance matrix

The program can be divided up so that the variance-covariance matrix is computed in a separate run. This may be useful on computers where storage is limited.

C. Single precision

The master deck containing the program will be the double precision version. The change to single precision is accomplished by the following steps.

1. Remove all double precision statements
2. Change double precision constants, which are all indicated by the D

format. Example: change 0.5D0 to 0.5

3. Change D's in format statements to E format

4. Change the names of the FORTRAN functions called to the single precision version, i.e.,

DABS	to	ABS
DSQRT	to	SQRT
DLOG	to	ALOG
DEXP	to	EXP

DLOG and DEXP do not occur in this package, but they may occur in the subroutine LOGLIK.

D. <u>Inclusion of different or additional iteration methods</u>

The subroutine package is designed so that it is simple to include different or additional iteration methods without changing the structure of the package; the control of options, input and output, computation of the variance-covariance matrix, and the test for convergence can remain the same.

VII. EXAMPLES

Given a set of n observations $y_1, y_2, \ldots, y_n$ which are a random sample from a mixture of k univariate normal distributions $N(\mu_1, \sigma^2)$, $N(\mu_2, \sigma^2), \ldots, N(\mu_k, \sigma^2)$, the following programs obtain the maximum likelihood estimates for the means $\mu_1, \mu_2, \ldots, \mu_k$, the common variance σ^2, and the proportions of admixture $p_1, p_2, \ldots, p_k$ of the k distributions. The log likelihood, L, is given by

$$L = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln \sigma^2 + \sum_{i=1}^{n} \ln \sum_{j=1}^{k} p_j e^{f_{ij}}$$

where

$$f_{ij} = -(y_i - \mu_j)^2 / 2\sigma^2 .$$

In example 1, only the essential sections of LOGLIK are used. The model and data are identical in example 2, but more of the program options are used. Also, the results in example 2 are obtained using the logarithms of the original data measurements, in order to illustrate the possible options concerning transformations. The results of this example include "negative variances"; example 3 shows how these are eliminated by a model in which fewer parameters are estimated.

The driver program and subroutines in these examples have been written in a general form, to illustrate the flexibility possible.

A. Example 1

```
C     THIS IS AN EXAMPLE OF THE PROGRAM THAT CALLS THE MAXIMUM
C     LIKELIHOOD PACKAGE, MAXLIK.
C
      DOUBLE PRECISION X,V
      EXTERNAL MKUNDL,MKUNDR
C
C     MKUNDL  IS THE NAME OF THE SUBROUTINE THAT COMPUTES THE LOG
C     LIKELIHOOD;  IT CORRESPONDS TO LOGLIK IN THE WRITE-UP.
C     MKUNDR IS THE NAME OF THE SUBROUTINE THAT COMPUTES THE
C     FUNCTIONALLY DEPENDENT PARAMETERS;
C     IT CORRESPONDS TO DEPAR IN THE WRITE-UP.
C
      DIMENSION X(11),V(121)
C
C     THIS PROGRAM ESTIMATES THE MEANS,PROPORTIONS OF ADMIXTURE, AND
C     COMMON VARIANCE FOR A MIXTURE OF K NORMAL DISTRIBUTIONS.
C     THERE ARE 2K+1 PARAMETERS. THESE DIMENSIONS ARE SUFFICIENT
C     FOR K UP TO 5.
C
    1 CALL MAXLIK(MKUNDL,MKUNDR,X,V)
C
C     RETURN TO MAXLIK TO DO ANOTHER ANALYSIS BY RETURNING TO
C     STATEMENT 1.
C     A BLANK CARD IN THE PROGRAM PARAMETERS WILL STOP THE RUN IN MAXLIK
C
      GO TO 1
      END

      SUBROUTINE MKUNDL(DL,X)
C
C     SUBROUTINE MKUNDL COMPUTES THE LIKELIHOOD GIVEN A SET OF PARAMETER
C     VALUES IN X AND STORES IT IN DL.
C
      DOUBLE PRECISION DL,X,XL,XU,FOPT,FEST,Z,XTK,XTK2,Y,PI,PI2
      DOUBLE PRECISION RN,RND2,CONST,SS,SS2,SUM,D
      COMMON/MLECAM/
     2 XL(75),XU(75),FOPT(10),FEST(75),Z(75),XTK,XTK2,
     3 IFR(75),LRX(75),IOPT(40),IDIST(20),LABEL(75,7),KTRAN,NT,NR,NS
C
C     THE DIMENSIONS THAT ARE 75 IN THE COMMON STATEMENT IN THIS EXAMPLE
C     MUST BE IDENTICAL TO THE DIMENSIONS COMPILED IN THE SAME COMMON
C     IN THE MAXIMUM LIKELIHOOD PACKAGE. (SEE  PP. IV-6 AND V-1.)
C
      COMMON/MKUND/KP2,KK
C
C     THE COMMON/MKUND/ STATEMENT IS FOR PASSING INFORMATION TO MKUNDR.
C
C
      DIMENSION Y(1000),    X(1)
C
C     THESE DIMENSIONS ALLOW SAMPLE SIZE UP TO 1000.
C
      LENTRY=IOPT(14)
      GO TO (1,2),LENTRY
C     SEE P. IV-7 FOR A DISCUSSION OF IOPT(14).
C
C     SECTION (T)
C
C     THE STATEMENTS BEGINNING WITH STATEMENT NO. 1 ARE ONLY EXECUTED
C     THE FIRST TIME THIS SUBROUTINE IS CALLED FOR EACH ANALYSIS.
C
    1 READ(1,100)N,K
  100 FORMAT(2I5)
C
C     N=SAMPLE SIZE
C     K=NO. OF DISTRIBUTIONS
C
      KPU=K+1
      KP2=K+2
      KK=K+K
```

```
C     THE CURRENT ESTIMATES ARE PASSED TO MKUNDL IN THE ARRAY X
C     THE ORDER OF THE PARAMETERS IS AS FOLLOWS:
C     X(1)=MEAN OF DISTRIBUTION 1
C     X(2)=MEAN OF DISTRIBUTION 2
C      .
C      .
C      .
C     X(K)=MEAN OF DISTRIBUTION K
C     X(K+1)=COMMON VARIANCE
C     X(K+2)=PROPORTION OF ADMIXTURE OF DISTRIBUTION 1
C     X(K+3)=PROPORTION OF ADMIXTURE OF DISTRIBUTION 2
C      .
C      .
C      .
C     X(2K+1)=PROPORTION OF ADMIXTURE OF DISTRIBUTION K
C
      KMU=K-1
      PI=3.1415926536D0
      PI2=PI+PI
      CALL RDATA(Y,N)
C
C     RDATA IS A USER SUPPLIED SUBROUTINE THAT READS THE DATA TO BE
C     ANALYZED INTO THE ARRAY Y. N IS THE SAMPLE SIZE.
C
      RN=N
      RND2=RN/2.D0
      CONST=-RND2*DLOG(PI2)
C
C
C     SECTION (II)
C
C     THE STATEMENTS BEGINNING WITH STATEMENT NO. 2 ARE EXECUTED EACH
C     TIME MKUNDL IS CALLED DURING THE COMPUTATION OF THE ML ESTIMATES
C     AND THE VARIANCE-COVARIANCE MATRIX.
C
    2 SS=X(K+1)
      SS2=SS+SS
C
C     SS IS THE COMMON VARIANCE.
C
C     IN THIS MODEL, ONE PROPORTION OF ADMIXTURE IS ALWAYS FUNCTIONALLY
C     DEPENDENT AND MUST BE COMPUTED IN MKUNDR, UNLESS ALL THE
C     PROPORTIONS ARE FIXED. IN THE LATTER CASE, NS=0 AND THE CALL TO
C     MKUNDR IS SKIPPED. NS IS READ IN MAXLIK, PARAMETER CARD 2, AND
C     STORED IN COMMON/MLECAM/.
C
      IF (NS.EQ.0) GO TO 10
      CALL MKUNDR(X,LEXIT)
C
      IF (LEXIT) 10,10,11
C
C     LEXIT GREATER THAN 0 INDICATES THAT THE DEPENDENT PARAMETER IS
C     OUTSIDE ITS BOUNDS SO THE LIKELIHOOD CANNOT BE EVALUATED.
C
C
   11 IOPT(11)=1
      RETURN
C
C     THE SUMMATION OVER ALL INDIVIDUALS IN THE SAMPLE BEGINS HERE.
C
   10 DL=0.D0
      DO 12 I=1,N
      SUM=0.D0
      DO 15 J=1,K
      D=Y(I)-X(J)
   15 SUM=SUM+X(KPU+J)*DEXP(-(D*D)/SS2)
      IF (SUM) 11,11,12
C
C     IF SUM IS LESS THAN OR EQUAL TO 0 , THE LIKELIHOOD IS NOT
C     DEFINED FOR THE CURRENT VALUES OF THE PARAMETERS.
```

```
   12 DL=DL+DLOG(SUM)
      DL=DL+CONST-RND2*DLOG(SS)
C
      IOPT(36)=IOPT(36)+1
C
C     IOPT(36) COUNTS THE NO. OF TIMES THE LIKELIHOOD IS COMPUTED.
C     THIS STATEMENT IS NOT ESSENTIAL.
C
      RETURN
      END


C     SUBROUTINE MKUNDR COMPUTES THE PROPORTION OF ADMIXTURE OF THE
C     K-TH DISTRIBUTION AS 1-THE SUM OF THE PROPORTIONS OF THE OTHER
C     K-1 DISTRIBUTIONS AND TESTS THAT IT IS BETWEEN 0 AND 1.
C
      SUBROUTINE MKUNDR(X,LEXIT)
      DOUBLE PRECISION X,PSUM,P
      DIMENSION X(1)
      COMMON/MKUND/KP2,KK
      LEXIT=0
      PSUM=0.D0
      DO 1 J=KP2,KK
    1 PSUM=PSUM+X(J)
      P=1.D0-PSUM
      X(KK+1)=P
      IF (P.LT.0.D0.OR.P.GT.1.D0) LEXIT=1
      RETURN
      END
C
C     RDATA IS A USER SUPPLIED SUBROUTINE THAT READS THE DATA
C
      SUBROUTINE RDATA(Y,N)
      DOUBLE PRECISION Y
      DIMENSION Y(1),LFMT(20)
      READ(1,1) KUNIT,LFMT
    1 FORMAT(I1/20A4)
      READ(KUNIT,LFMT) (Y(I),I=1,N)
      RETURN
      END
```

THE FOLLOWING IS A LISTING OF THE DATA INPUT CARDS, (INCLUDING BOTH PROGRAM PARAMETERS AND THE DATA TO BE ANALYZED), FOR EXAMPLE 1 .

```
RED CELL SODIUM DATA. SAMPLE 3 MALES.
0.001                 0.5                 50  7  7  1 110 102  1100                11
  7  2  5  6
7.490     11.650    15.107    2.135     0.7426    0.1548    0.1026
0.0       0.0       0.0       0.001     0.0       0.0       0.0
100.0     100.0     100.0     100.0     1.0       1.0       1.0
   47    3
1
(4X,D4.1)
    09.9
    11.7
    15.5
    09.9
    08.2
    09.7
    12.9
    09.9
    11.0
    05.8
    06.8
    06.5
    13.9
    04.5
    08.2
    13.0
    13.1
    15.1
    07.5
    08.7
    06.7
    07.8
    09.2
    06.1
    03.6
    06.8
    06.2
    05.5
    07.1
    12.3
    11.1
    06.3
    07.6
    06.3
    08.9
    06.8
    07.0
    08.5
    08.9
    07.9
    17.6
    11.9
    06.8
    08.9
    11.7
    08.3
    07.6
         CARD WITH BLANKS IN AT LEAST COLUMNS 1 THROUGH 4 INDICATES END OF RUN.
```

Output

```
RED CELL SODIUM DATA. SAMPLE 3 MALES.

PROGRAM PARAMETERS READ IN

ITERATION PARAMETER CARD

 0.0010000 0.0        0.5000 0    0 0 50   7   7   1 110 102   1100        0.0         0.0    130000    0

PARAMETER   7 IS FUNCTIONALLY DEPENDENT ON THE   2 PARAMETERS   5   6

INITIAL ESTIMATES

      7.49000       11.65000       15.10700       2.13500       0.74260       0.15480       0.10260

LOWER BOUNDS

      0.0           0.0            0.0            0.00100       0.0           0.0           0.0

UPPER BOUNDS

    100.00000     100.00000      100.00000      100.00000       1.00000       1.00000       1.00000

PROGRAM PARAMETERS AFTER FIRST CALL TO USERS LIKELIHOOD SUBROUTINE

   LOWER BOUNDS   INITIAL ESTIMATES   UPPER BOUNDS IRF LRX

 1      0.0             7.490000    100.000000      0   0

 2      0.0            11.650000    100.000000      0   0

 3      0.0            15.107000    100.000000      0   0

 4      0.001000        2.135000    100.000000      0   0

 5      0.0             0.742600      1.000000      1   0

 6      0.0             0.154800      1.000000      1   0

 7      0.0             0.102600      1.000000     -1   1

LIKELIHOOD OF INITIAL ESTIMATES    -0.1142414674D 03

DIRECT SEARCH OF LIKELIHOOD SURFACE

ITERATION    1 OLD             NEW            STEP SIZE    ABSOLUTE ERROR
 1        7.49000000       7.45829256      0.03337784       0.031707[illegible]
 2       11.65000000      11.90000000      0.02145923       0.25000000
 3       15.10700000      15.03417863      0.01654862       0.07282[illegible]37
 4        2.13500000       2.64127539      0.50000000       0.50627539
 5        0.74260000       0.74817741      0.02550341       0.00557741
 6        0.15480000       0.17865214      0.06250000       0.02385[illegible]
 7        0.10260000       0.07317045      0.50000000       0.02942955

OLD LOG LIKELIHOOD      -114.2414674445 NEW LOG LIKELIHOOD      -113.8544310491

MAXIMUM ERROR        0.50627539 CHANGE IN LOG LIKELIHOOD        0.3870363955

NO. OF LIKELIHOOD EVALUATIONS     31
```

RESULTS OF 14 INTERMEDIATE ITERATIONS OMITTED HERE

ITERATION 16 OLD		NEW	STEP SIZE	ABSOLUTE ERROR
1	7.43938050	7.43966067	0.00100000	0.00028018
2	12.06485303	12.06607707	0.00100000	0.00122404
3	15.94052327	15.94216985	0.00100000	0.00164658
4	2.28872804	2.28907709	0.00100000	0.00034905
5	0.69984491	0.69985961	0.00100000	0.00001471
6	0.24386439	0.24389638	0.00024357	0.00003199
7	0.05629070	0.05624400	0.00011324	0.00004670

OLD LOG LIKELIHOOD -113.3874805950 NEW LOG LIKELIHOOD -113.3874768761

MAXIMUM ERROR 0.00164658 CHANGE IN LOG LIKELIHOOD 0.0000037189

NO. OF LIKELIHOOD EVALUATIONS 293

CONVERGENCE IN 16 ITERATIONS

VARIANCE-COVARIANCE MATRIX
BEFORE ADDING ROWS AND COLUMNS FOR PARAMETERS THAT ARE FUNCTIONALLY DEPENDENT OR HAVE CONVERGED TO A BOUND

0.110878	0.127815	0.072473	0.066427	0.010622	-0.006752
0.127815	0.847471	0.632215	0.178315	0.037382	-0.008983
0.072473	0.632215	2.225198	0.099750	0.022047	0.024233
0.066427	0.178315	0.099750	0.464621	0.013109	-0.004087
0.010622	0.037382	0.022047	0.013109	0.007392	-0.005454
-0.006752	-0.008983	0.024233	-0.004087	-0.005454	0.006490

INVERSE CHECK

1.000000	-0.000000	-0.000000	-0.000000	-0.000000	0.000000
-0.000000	1.000000	-0.000000	-0.000000	-0.000000	0.0
0.0	-0.000000	1.000000	-0.000000	0.000000	-0.000000
0.000000	0.000000	-0.000000	1.000000	0.000000	-0.000000
0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
0.000000	-0.000000	0.000000	0.000000	-0.000000	1.000000

NEWTON-RAPHSON METHOD
WITHOUT RECALCULATING VARIANCE-COVARIANCE MATRIX

ITERATION 17 OLD		NEW	STEP SIZE	ABSOLUTE ERROR
1	7.43966067	7.43988537	0.00100000	0.00022470
2	12.06607707	12.06698139	0.00100000	0.00090432
3	15.94216985	15.94356772	0.00100000	0.00139787
4	2.28907709	2.28934354	0.00100000	0.00026646
5	0.69985961	0.69981180	0.00100000	0.00004782
6	0.24389638	0.24397291	0.00100000	0.00007652
7	0.05648790	0.05621530	0.00100000	0.00027260

OLD LOG LIKELIHOOD -113.3874768761 NEW LOG LIKELIHOOD -113.3874754291

MAXIMUM ERROR 0.00139787 CHANGE IN LOG LIKELIHOOD 0.0000014471

NO. OF LIKELIHOOD EVALUATIONS 616

SUMMARY OF RESULTS

MAXIMUM LIKELIHOOD ESTIMATES FOR

RED CELL SODIUM DATA. SAMPLE 3 MALES.

ML PROGRAM APR. 10, 1973 VERSION

DIRECT SEARCH OF LIKELIHOOD SURFACE

PARAMETER	LOWER BOUNDS	INITIAL ESTIMATES	UPPER BOUNDS
	0.0	7.49000	100.00000
	0.0	11.65000	100.00000
	0.0	15.10700	100.00000
	0.00100	2.13500	100.00000
	0.0	0.74260	1.00000
	0.0	0.15480	1.00000
	0.0	0.10260	1.00000

LIKELIHOOD OF INITIAL ESTIMATES -0.1142414674D 03

MAXIMUM RELATIVE ESTIMATE CHANGE REQUESTED 0.00100 USED 0.00100

INITIAL STEP SIZE 0.50000

THE FOLLOWING PARAMETERS ARE FUNCTIONALLY DEPENDENT
7

PARAMETER	FINAL ESTIMATES	STD. ERROR
	7.43989	0.33298
	12.06698	0.92058
	15.94357	1.49171
	2.28934	0.68163
	0.69981	0.08598
	0.24397	0.08056
	0.05622	0.05452

JOINT LOG LIKELIHOOD OF THE SAMPLE = -0.1133874754D 03

MAXIMUM LAST ITERATION CHANGE IN ESTIMATES 0.0013979

CONVERGENCE IN 17 ITERATIONS

RESULTS VERIFIED BY SEARCH AROUND FINAL ESTIMATES

FINAL ESTIMATES IMPROVED BY ONE NEWTON-RAPHSON ITERATION

VARIANCE-COVARIANCE MATRIX IS BASED ON ESTIMATES OBTAINED BY DIRECT SEARCH WHICH ARE:

7.43966

12.06608

15.94217

2.28908

0.69986

0.24390

0.05649

AT END OF DIRECT SEARCH LIKELIHOOD WAS -0.1133874769D 03 . MAXIMUM PREVIOUS ITERATION CHANGE 0.0016466

VARIANCE-COVARIANCE MATRIX

0.110878	0.127815	0.072473	0.066427	0.010622	-0.006752	-0.003870
0.127815	0.847471	0.632215	0.178315	0.037382	-0.008983	-0.028398
0.072473	0.632215	2.225198	0.099750	0.022047	0.024233	-0.046279
0.066427	0.178315	0.099750	0.464621	0.013109	-0.004087	-0.009022
0.010622	0.037382	0.022047	0.013109	0.007392	-0.005454	-0.001938
-0.006752	-0.008983	0.024233	-0.004087	-0.005454	0.006490	-0.001035
-0.003870	-0.028398	-0.046279	-0.009022	-0.001938	-0.001035	0.002973

FIRST DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
WITH RESPECT TO INDEPENDENT PARAMETERS NOT FIXED

0.001868	0.001182	0.000384	0.000240	-0.016596	0.000141

NEGATIVES OF THE SECOND DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
WITH RESPECT TO INDEPENDENT PARAMETERS NOT FIXED

11.959020	-1.379068	-0.006768	-1.034652	-2.750509	7.595701
-1.379068	2.267836	-0.310517	-0.247021	-16.122993	-10.842871
-0.006768	-0.310517	0.716308	0.103310	-7.865318	-9.657344
-1.034652	-0.247021	0.103310	2.489209	-5.689137	-5.018113
-2.750509	-16.122993	-7.865318	-5.689137	670.700576	564.325376
7.595701	-10.842871	-9.657344	-5.018113	564.325376	654.199477

NO. OF LIKELIHOOD EVALUATIONS 617

(These computations are performed automatically since the first Newton-Raphson iteration at the end of the direct search gave improved results.)

NEWTON-RAPHSON METHOD
WITHOUT RECALCULATING VARIANCE-COVARIANCE MATRIX

ITERATION	1 OLD	NEW	STEP SIZE	ABSOLUTE ERROR
1	7.43988537	7.43988549	0.00100000	0.00000012
2	12.06698139	12.06698197	0.00100000	0.00000059
3	15.94356772	15.94356886	0.00100000	0.00000114
4	2.28934354	2.28934401	0.00100000	0.00000047
5	0.69981180	0.69981182	0.00100000	0.00000002
6	0.24397291	0.24397293	0.00100000	0.00000002
7	0.05621530	0.05621526	0.00100000	0.00000004

OLD LOG LIKELIHOOD -113.3874754291 NEW LOG LIKELIHOOD -113.3874754290

MAXIMUM ERROR 0.00000114 CHANGE IN LOG LIKELIHOOD 0.0000000000

NO. OF LIKELIHOOD EVALUATIONS 738

CONVERGENCE IN 1 ITERATIONS

VARIANCE-COVARIANCE MATRIX
BEFORE ADDING ROWS AND COLUMNS FOR PARAMETERS THAT ARE FUNCTIONALLY DEPENDENT OR HAVE CONVERGED TO A BOUND

0.110902	0.127941	0.072727	0.066523	0.010627	-0.006749
0.127941	0.848177	0.633520	0.178850	0.037413	-0.008972
0.072727	0.633520	2.228026	0.100774	0.022110	0.024243
0.066523	0.178850	0.100774	0.464906	0.013129	-0.004074
0.010627	0.037413	0.022110	0.013129	0.007395	-0.005455
-0.006749	-0.008972	0.024243	-0.004074	-0.005455	0.006491

INVERSE CHECK

1.000000	-0.000000	-0.000000	-0.000000	-0.000000	0.000000
-0.000000	1.000000	-0.000000	-0.000000	-0.000000	0.0
0.000000	0.000000	1.000000	0.000000	0.000000	-0.000000
0.000000	-0.000000	-0.000000	1.000000	0.000000	-0.000000
-0.000000	-0.000000	0.000000	-0.000000	1.000000	0.000000
0.000000	0.000000	0.000000	0.000000	-0.000000	1.000000

SUMMARY OF RESULTS

MAXIMUM LIKELIHOOD ESTIMATES FOR

RED CELL SODIUM DATA. SAMPLE 3 MALES.

ML PROGRAM APR. 10, 1973 VERSION

NEWTON-RAPHSON METHOD
WITHOUT RECALCULATING VARIANCE-COVARIANCE MATRIX

PARAMETER	LOWER BOUNDS	INITIAL ESTIMATES	UPPER BOUNDS
	0.0	7.43989	100.00000
	0.0	12.06698	100.00000
	0.0	15.94357	100.00000
	0.00100	2.28934	100.00000
	0.0	0.69981	1.00000
	0.0	0.24397	1.00000
	0.0	0.05622	1.00000

LIKELIHOOD OF INITIAL ESTIMATES -0.1133874754D 03

MAXIMUM RELATIVE ESTIMATE CHANGE REQUESTED 0.00100 USED 0.00100

THE FOLLOWING PARAMETERS ARE FUNCTIONALLY DEPENDENT
7

PARAMETER	FINAL ESTIMATES	STD. ERROR
	7.43989	0.33302
	12.06698	0.92097
	15.94357	1.49266
	2.28934	0.68184
	0.69981	0.08599
	0.24397	0.08057
	0.05622	0.05454

JOINT LOG LIKELIHOOD OF THE SAMPLE = -0.1133874754D 03

MAXIMUM LAST ITERATION CHANGE IN ESTIMATES 0.0000011

CONVERGENCE IN 1 ITERATIONS

VARIANCE-COVARIANCE MATRIX

0.110902	0.127941	0.072727	0.066523	0.010627	-0.006749	-0.003878
0.127941	0.848177	0.633520	0.178850	0.037413	-0.008972	-0.028441
0.072727	0.633520	2.228026	0.100774	0.022110	0.024243	-0.046353
0.066523	0.178850	0.100774	0.464906	0.013129	-0.004074	-0.009055
0.010627	0.037413	0.022110	0.013129	0.007395	-0.005455	-0.001939
-0.006749	-0.008972	0.024243	-0.004074	-0.005455	0.006491	-0.001036
-0.003878	-0.028441	-0.046353	-0.009055	-0.001939	-0.001036	0.002975

FIRST DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
WITH RESPECT TO INDEPENDENT PARAMETERS NOT FIXED

0.0	0.0	0.000000	0.000001	-0.000091	0.0

NEGATIVES OF THE SECOND DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
WITH RESPECT TO INDEPENDENT PARAMETERS NOT FIXED

11.958510	-1.378959	-0.006753	-1.035000	-2.749972	7.591765
-1.378959	2.267767	-0.310513	-0.247386	-16.128552	-10.849634
-0.006753	-0.310513	0.715781	0.102752	-7.864140	-9.654450
-1.035000	-0.247386	0.102752	2.488742	-5.694449	-5.025601
-2.749972	-16.128552	-7.864140	-5.694449	670.773051	564.387549
7.591765	-10.849634	-9.654450	-5.025601	564.387549	654.198013

NO. OF LIKELIHOOD EVALUATIONS 1001

B. Example 2

```
C     SUBROUTINE MKUNDL COMPUTES THE LIKELIHOOD GIVEN A SET OF PARAMETER
C     VALUES IN X AND STORES IT IN DL.
C
      SUBROUTINE MKUNDL(DL,X)
      DOUBLE PRECISION DL,X,XL,XU,FOPT,FEST,Z,XTK,XTK2,Y,PI,PI2
      DOUBLE PRECISION RN,RND2,CONST,SS,SS2 SUM,D
      DOUBLE PRECISION EPIJ,PREVX
      DOUBLE PRECISION WMEAN,WW
      COMMON/MLECAM/
     * XL(75),XU(75),FOPT(10),FEST(75),Z(75),XTK,XTK2,
     * IFR(75),LRX(75),IOPT(40),IDIST(20),LABEL(75,7),KTRAN,NT,NR,NS
C
C     THE DIMENSIONS THAT ARE 75 IN THE COMMON STATEMENT IN THIS EXAMPLE
C     MUST BE IDENTICAL TO THE DIMENSIONS COMPILED IN THE SAME COMMON
C     IN THE MAXIMUM LIKELIHOOD PACKAGE. (SEE  PP. IV-6 AND V-1.)
C
      COMMON/MKUND/KP2,KK
C
C     THE COMMON/MKUND/ STATEMENT IS FOR PASSING INFORMATION TO MKUNDR.
C
      DIMENSION Y(1000),     X(1)
      DIMENSION EPIJ(1000,5),PREVX(11)
C
C     THESE DIMENSIONS ALLOW SAMPLE SIZE UP TO 1000.
C
      DIMENSION JDIST(20),LMU(5),LSS(4),LP(6),LK(5)
C
C     THE ABOVE ARRAYS STORE DATA FOR MODEL AND PARAMETER LABELS
C
      DATA JDIST
     2 / 4HMIXT,4HURE ,4H OF ,4HUNIV,4HARIA,4HTE N, 4HORMA,4HL DI,
     3   4HSTRI,4HB.  /
      DATA LMU
     2 / 4HMEAN,4H OF ,4HDIST,4HRIBU,4HTION/
      DATA LSS
     2 / 4HCOMM,4HON V,4HARIA,4HNCE /
      DATA LP
     2 / 4HPROP,4HORTI,4HON O,4HF AD,4HMIXT,4HURE /
      DATA LK /4H 1  ,4H 2  ,4H 3  ,4H 4  ,4H 5  /
C
      LENTRY=IOPT(14)
      GO TO (1,2,3,4,5,6,7),LENTRY
C
C     ALL 7 SECTIONS OF MKUNDL ARE INCLUDED IN THIS EXAMPLE.
C     SEE P. IV-7 FOR A DISCUSSION OF IOPT(14).
C
C
C     SECTION  (I)
C
C     THE STATEMENTS BEGINNING WITH STATEMENT NO. 1 ARE ONLY EXECUTED
C     THE FIRST TIME THIS SUBROUTINE IS CALLED FOR EACH ANALYSIS.
C
    1 READ(1,100)N,K
  100 FORMAT(2I5)
C
C     N=SAMPLE SIZE
C     K=NO. OF DISTRIBUTIONS
C
      KPU=K+1
      KP2=K+2
      KK=K+K
      KX=KK+1
      KMU=K-1
C
C     THE CURRENT ESTIMATES ARE PASSED TO MKUNDL IN THE ARRAY X
C     THE ORDER OF THE PARAMETERS IS AS FOLLOWS:
C     X(1)=MEAN OF DISTRIBUTION 1
C     X(2)=MEAN OF DISTRIBUTION 2
        .
        .
        .
```

```
C     X(K)=MEAN OF DISTRIBUTION K
C     X(K+1)=COMMON VARIANCE
C     X(K+2)=PROPORTION OF ADMIXTURE OF DISTRIBUT:(N 1
C     X(K+3)=PROPORTION OF ADMIXTURE OF DISTRIBUT:(N 2
C      .
C      .
C      .
C     X(2K+1)=PROPORTION OF ADMIXTURE OF DISTRIBU!ION K
C
      PI=3.1415926536D0
      PI2=PI+PI
      IF (IOPT(1).GT.0) GO TO 9
C
C     IOPT(1) IS SET IN MAXLIK TO THE VALUE KRDAT READ IN ON PARAMETER
C     CARD 2. IT IS USED HERE TO SKIP READING OF (ATA IF IT IS GREATER
C     THAN 0.
C
C
      CALL RTRDAT(Y,N,KTRAN,XTK,XTK2)
C     RTRDAT IS A USER SUPPLIED SUBROUTINE TO REA( AND TRANSFORM DATA.
C     KTRAN,XTK,AND XTK2 ARE TRANSFORMATION CODES READ IN MAXLIK ON
C     PARAMETER CARD 2 AND STORED IN COMMON/MLECAP/.
C     XTK2 IS NOT USED IN THIS EXAMPLE.
C     THE TRANSFORMED DATA IS STORED IN Y IN RTRDAT.
C     N IS THE SAMPLE SIZE.
C
    9 RN=N
      RND2=RN/2.D0
      CONST=-RND2*DLOG(PI2)
      DO 8 J=1,KX
    8 PREVX(J)=99999999.D0
C
C     PREVX WILL CONTAIN THE VALUES OF THE PARAMETERS FOR THE PREVIOUS
C     LIKELIHOOD COMPUTATION IN PART 2 OF MKUNDL. I( PART 1 IT IS
C     INITIALIZED TO ANY 'IMPOSSIBLE' VALUES.
C
      IF (IOPT(2).EQ.0.OR.KTRAN.LT.2) GO TO 13
C
C     IOPT(2) IS SET IN MAXLIK TO THE VALUE KREST READ IN ON PARAMETER CARD 2.
C     IF IOPT(2) IS 0, INITIAL ESTIMATES WERE NOT READ IN; AND IF
C     KTRAN IS LESS THAN 2, THE DATA ARE NOT TRANSFORMED; IN EITHER
C     CASE, THE INITIAL ESTIMATES SHOULD NOT BE TRANSFORMED.
C
C     OTHERWISE TRANSFORM THE INITIAL ESTIMATES AND BOUNDS.
C     WHEN USING THIS SUBROUTINE,
C     INITIAL ESTIMATE AND BOUNDS MUST BE READ IN I( THE ORIGINAL SCALE.
C
      WMEAN=0.D0
      DO 25 J=1,K
      JJ=J+K+1
      Z(J)=DLOG(X(J)+XTK)
      XU(J)=DLOG(XU(J)+XTK)
      IF (XL(J).LE.0) GO TO 26
      XL(J)=DLOG(XL(J)+XTK)
C
C     XTK MUST BE CHOSEN SUCH THAT ALL OBSERVATIONS AND THE BOUNDS ARE
C     POSITIVE WHEN XTK IS ADDED.
C
   27 WMEAN=WMEAN+(X(J)*X(JJ))
   25 X(J)=Z(J)
      WW=WMEAN*WMEAN
      X(K+1)=X(K+1)/WW
      XL(K+1)=XL(K+1)/WW
      XU(K+1)=XU(K+1)/WW
C
   13 CONTINUE
C
C     STORE NAME OF MODEL IN IDIST.
      DO 21 J=1,10
   21 IDIST(J)=JDIST(J)
```

```
C     STORE NAMES OF INDIVIDUAL PARAMETERS IN LABEL
C
      DO 22 I=1,K
      LABEL(I,6)=LK(I)
      DO 22 J=1,5
   22 LABEL(I,J)=LMU(J)
      DO 23 J=1,4
   23 LABEL(K+1,J)=LSS(J)
      DO 24 I=KP2,NT
      II=I-K-1
      LABEL(I,7)=LK(II)
      DO 24 J=1,6
   24 LABEL(I,J)=LP(J)
C
C
C     SECTION (II)
C
C
C     THE STATEMENTS BEGINNING WITH STATEMENT NO. 2 ARE EXECUTED EACH
C     TIME MKUNDL IS CALLED DURING THE COMPUTATION OF THE ML ESTIMATES
C     AND THE VARIANCE-COVARIANCE MATRIX.
C
    2 SS=X(K+1)
      SS2=SS+SS
C
C     SS IS THE COMMON VARIANCE.
C
C     IN THIS MODEL, ONE PROPORTION OF ADMIXTURE IS ALWAYS FUNCTIONALLY
C     DEPENDENT AND MUST BE COMPUTED IN MKUNDR, UNLESS ALL THE
C     PROPROTIONS ARE FIXED. IN THE LATTER CASE, NS=0 AND THE CALL TO
C     MKUNDR IS SKIPPED. NS IS READ IN MAXLIK, PARAMETER CARD 2, AND
C     STORED IN COMMON/MLECAM/.
C
      IF (NS.EQ.0) GO TO 10
      CALL MKUNDR(X,LEXIT)
C
      IF (LEXIT) 10,10,11
C
C     LEXIT GREATER THAN 0 INDICATES THAT THE DEPENDENT PARAMETER IS
C     OUTSIDE ITS BOUNDS SO THE LIKELIHOOD CANNOT BE EVALUATED.
C
   11 IOPT(11)=1
      RETURN
C
C     LOWER BOUND FOR MEANS  WHEN LOG OF THE DATA ARE ANALYZED.
   26 XL(J)=-10000.D0
      GO TO 27
C
C
C     THE SUMMATION OVER ALL INDIVIDUALS IN THE SAMPLE BEGINS HERE.
C
   10 DL=0.D0
      DO 14 J=1,KPU
      IF (X(J).NE.PREVX(J)) GO TO 16
   14 CONTINUE
C
C     IF VARIANCE AND ALL MEANS ARE SAME AS FOR PREVIOUS LIKELIHOOD
C     COMPUTATION USE PREVIOUSLY COMPUTED VALUES AND GO TO 17.
C
      GO TO 17
   16 DO 12 J=1,K
C
      IF (X(J).EQ.PREVX(J).AND.SS.EQ.PREVX(KPU)) GO TO 12
C
C     IF THE MEAN OF THE J-TH DISTRIBUTION AND THE VARIANCE ARE BOTH
C     UNCHANGED FROM THE PREVIOUS LIKELIHOOD COMPUTATION, GO TO 12.
```

```
      DO 15 I=1,N
      D=Y(I)-X(J)
      EFIJ(I,J)=DEXP(-D*D/SS2)
   15 CONTINUE
   12 CONTINUE
C
      DO 29 J=1,KPU
   29 PREVX(J)=X(J)
C
C     STORE CURRENT VALUES OF PARAMETERS IN PREVX
C
   17 DO 18 I=1,N
      SUM=0.D0
      DO 19 J=1,K
   19 SUM=SUM+X(KPU+J)*EFIJ(I,J)
      IF (SUM)11,11,18
C
C     IF SUM IS LESS THAN OR EQUAL TO 0 , THE LIKELIHOOD IS NOT
C     DEFINED FOR THE CURRENT VALUES OF THE PARAMETERS.
C
   18 DL=DL+DLOG(SUM)
      DL=DL+CONST-RND2*DLOG(SS)
C
      IOPT(36)=IOPT(36)+1
C
C     IOPT(36) COUNTS THE NO. OF TIMES THE LIKELIHOOD IS COMPUTED.
C     THIS STATEMENT IS NOT ESSENTIAL.
C
      RETURN
C
C     SECTION (III)
C
C
C     IF LOGS OF THE DATA WERE ANALYZED TRANSFORM THE FINAL ESTIMATES
C
C     TO THE ORIGINAL SCALE AND STORE IN Z
    3 WMEAN=0.D0
      DO 30 J=1,K
      JJ=J+K+1
      Z(JJ)=X(JJ)
C
C     THE PROPORTIONS ARE UNCHANGED
C
      Z(J)=DEXP(X(J))-XTK
   30 WMEAN=WMEAN+Z(J)*X(JJ)
C     WMEAN IS THE MEAN OF THE DISTRIBUTION MEANS IN ORIGINAL SCALE
C
      Z(K+1)=X(K+1)*WMEAN*WMEAN
C
C     SECTION (IV)
C
    4 CALL PRDATA(Y,N)
      RETURN
C
C     PRDATA IS A USER SUPPLIED SUBROUTINE TO PRINT THE DATA.
C
C
C     SECTION (V)
C
    5 CONTINUE
C     INSERT CODING HERE TO PERFORM ANY DESIRED COMPUTATIONS USING
C     THE FINAL ESTIMATES.
C
      RETURN
```

```
C
C
C     SECTION (VI)
C
    6 CONTINUE
C     INSERT CODING HERE TO PRINT THE RESULTS OF ANY COMPUTATIONS
C     PERFORMED IN SECTION (V).
C     THIS CODING SHOULD REPLACE THE FOLLOWING TWO STATEMENTS.
C
      WRITE(3,50)
   50 FORMAT(51HOANY OUTPUT PROGRAMMED IN SECTION (VI) OF THE USER ,
     * 47HSUPPLIED LIKELIHOOD SUBROUTINE IS PRINTED HERE.)
C
      RETURN
C
C
C     SECTION (VII)
C
C     ADDITIONAL IDENTIFYING OUTPUT
C
    7 WRITE(3,20)K,KTRAN,XTK,XTK2
   20 FORMAT(21H NO. OF DISTRIBUTIONS,I3/21H TRANSFORMATION CODES,I3,
     * 2F10.5)
      RETURN
      END
```

SUBROUTINE MKUNDR IS THE SAME AS FOR EXAMPLE 1.

```
C   RTRDAT IS A USER SUPPLIED SUBROUTINE TO READ AND TRANSFORM DATA.
C
      SUBROUTINE RTRDAT(Y,N,KTRAN,XTK,XTK2)
      DOUBLE PRECISION Y,XTK,XTK2
      DIMENSION Y(1),LFMT(20)
      READ(1,1)KUNIT,LFMT
    1 FORMAT(I1/20A4)
      READ(KUNIT,LFMT)(Y(I),I=1,N)
C
C     IF KTRAN IS GREATER THAN 1, LOGS OF THE DATA WILL BE TAKEN;
C     OTHERWISE THE DATA WILL REMAIN UNTRANSFORMED.
C
      IF (KTRAN.LE.1) RETURN
      DO 2 J=1,N
    2 Y(J)=DLOG(Y(J)+XTK)
      RETURN
      END
C     SUBROUTINE PRDATA IS A USER SUPPLIED SUBROUTINE TO PRINT THE DATA.
C
      SUBROUTINE PRDATA(Y,N)
      DOUBLE PRECISION Y
      DIMENSION Y(1)
      WRITE(3,1)(Y(J),J=1,N)
    1 FORMAT (1H0,8F11.5)
      RETURN
      END
```

THE DATA INPUT CARDS FOR EXAMPLE 2 ARE IDENTICAL TO THOSE FOR EXAMPLE 1 EXCEPT FOR THE ITERATION PARAMETER CARD, WHICH IS AS FOLLOWS:

```
0.001                 0.5             50  7  7  1 111 102  1102             13  11
```

THE ONLY DIFFERENCES ARE THAT:

1. THE DATA WILL BE TRANSFORMED. (COLUMN 59)
2. THE DATA WILL BE PRINTED. (COLUMN 49)
3. SECTIONS (V) AND (VI) OF THE LIKELIHOOD SUBROUTINE WILL BE USED. (COLUMN 76)
4. SECTION (VII) OF THE LIKELIHOOD SUBROUTINE WILL BE USED. (COLUMN 75)

Output

```
RED CELL SODIUM DATA. SAMPLE 3 MALES.

PROGRAM PARAMETERS READ IN

ITERATION PARAMETER CARD

 0.0010000 0.0        0.5000 0    0 0 50   7   7   1 111 102   1102        0.0          0.0   130011    0

PARAMETER   7 IS FUNCTIONALLY DEPENDENT ON THE   2 PARAMETERS   5   6

INITIAL ESTIMATES

      7.49000       11.65000      15.10700       2.13500       0.74260       0.15480       0.10260

LOWER BOUNDS

      0.0           0.0           0.0            0.00100       0.0           0.0           0.0

UPPER BOUNDS

    100.00000     100.00000     100.00000     100.00000      1.00000       1.00000       1.00000

PROGRAM PARAMETERS AFTER FIRST CALL TO USERS LIKELIHOOD SUBROUTINE

   LOWER BOUNDS    INITIAL ESTIMATES    UPPER BOUNDS IRP LRX

 1 -10000.000000        2.013569       4.605170      0  0

 2 -10000.000000        2.455306       4.605170      0  0

 3 -10000.000000        2.715158       4.605170      0  0

 4       0.000013       0.026860       1.258089      0  0

 5       0.0            0.742600       1.000000      1  0

 6       0.0            0.154800       1.000000      1  0

 7       0.0            0.102600       1.000000     -1  1

 LIKELIHOOD OF INITIAL ESTIMATES    -0.1984396106D 02


 DIRECT SEARCH OF LIKELIHOOD SURFACE
```

INTERMEDIATE OUTPUT OMITTED HERE.

CONVERGENCE IN 18 ITERATIONS.

```
VARIANCE-COVARIANCE MATRIX
BEFORE ADDING ROWS AND COLUMNS FOR PARAMETERS THAT ARE FUNCTIONALLY DEPENDENT OR HAVE CONVERGED
                                                                                        TO A BOUND


      0.004597     -0.002278      0.000585      0.001242     -0.000136    -24.120590
     -0.002278      0.021071     -0.034386     -0.002228      0.002320    -31.250745
      0.000585     -0.034386      0.458222     -0.002058      0.011067    -12.132232
      0.001242     -0.002228     -0.002058      0.000877     -0.000041     -1.429201
     -0.000136      0.002320      0.011067     -0.000041     -0.000250    -73.672940
    -24.120590    -31.250745    -12.132232     -1.429201    -73.672940*************
```

INVERSE CHECK

1.000000	0.000000	-0.000000	-0.000000	0.000000	0.000000
0.000000	1.000000	0.000000	0.000000	-0.000000	0.000000
0.0	-0.000000	1.000000	-0.000000	-0.000000	-0.000000
0.000000	0.000000	-0.000000	1.000000	-0.000000	-0.000000
0.000000	0.000000	0.000000	0.000000	1.000000	-0.000000
-0.000000	-0.000000	-0.000000	-0.000000	-0.000000	1.000000

NEWTON-RAPHSON METHOD
WITHOUT RECALCULATING VARIANCE-COVARIANCE MATRIX

ITERATION	19 OLD	NEW	STEP SIZE	ABSOLUTE ERROR
1	1.99338393	1.99339496	0.00100000	0.00001104
2	2.44324514	2.44322387	0.00100000	0.00002127
3	2.44322933	2.44322856	0.00100000	0.00000077
4	0.06085331	0.06086100	0.00100000	0.00000769
5	0.65343262	0.65343204	0.00100000	0.00000059
6	0.30930510	0.26980242	0.00100000	0.03950268
7	0.03757158	0.07676554	0.00100000	0.03919396

OLD LOG LIKELIHOOD -13.9508260679 NEW LOG LIKELIHOOD -13.9508260939

MAXIMUM ERROR 0.03950268 CHANGE IN LOG LIKELIHOOD 0.0000000260

NO. OF LIKELIHOOD EVALUATIONS 622

DATA

RED CELL SODIUM DATA. SAMPLE 3 MALES.

2.29253	2.45959	2.74084	2.29253	2.10413	2.27213	2.55723	2.29253
2.39790	1.75786	1.91692	1.87180	2.63189	1.50408	2.10413	2.56495
2.57261	2.71469	2.01490	2.16332	1.90211	2.05412	2.21920	1.80829
1.28093	1.91692	1.82455	1.70475	1.96009	2.50960	2.40695	1.84055
2.02815	1.84055	2.18605	1.91692	1.94591	2.14007	2.18605	2.06686
2.86790	2.47654	1.91692	2.18605	2.45959	2.11626	2.02815	

SUMMARY OF RESULTS

MAXIMUM LIKELIHOOD ESTIMATES FOR

MIXTURE OF UNIVARIATE NORMAL DISTRIB.

RED CELL SODIUM DATA. SAMPLE 3 MALES.
NO. OF DISTRIBUTIONS 3
TRANSFORMATION CODES 2 0.0 0.0

ML PROGRAM APR. 10, 1973 VERSION

DIRECT SEARCH OF LIKELIHOOD SURFACE

PARAMETER	LOWER BOUNDS	INITIAL ESTIMATES	UPPER BOUNDS
MEAN OF DISTRIBUTION 1	-10000.00000	2.01357	4.60517
MEAN OF DISTRIBUTION 2	-10000.00000	2.45531	4.60517
MEAN OF DISTRIBUTION 3	-10000.00000	2.71516	4.60517
COMMON VARIANCE	0.00001	0.02686	1.25809
PROPORTION OF ADMIXTURE 1	0.0	0.74260	1.00000
PROPORTION OF ADMIXTURE 2	0.0	0.15480	1.00000
PROPORTION OF ADMIXTURE 3	0.0	0.10260	1.00000

LIKELIHOOD OF INITIAL ESTIMATES -0.1984306106D 02

MAXIMUM RELATIVE ESTIMATE CHANGE REQUESTED 0.00100 USED 0.01000

INITIAL STEP SIZE 0.50000

THE FOLLOWING PARAMETERS ARE FUNCTIONALLY DEPENDENT
7

PARAMETER	FINAL ESTIMATES	STD. ERROR	APPROXIMATE VALUE IN ORIGINAL SCALE
MEAN OF DISTRIBUTION 1	1.99338	0.06780	7.34033
MEAN OF DISTRIBUTION 2	2.44325	0.14516	11.51033
MEAN OF DISTRIBUTION 3	2.44323	0.67692	11.51015
COMMON VARIANCE	0.06085	0.02961	4.70078
PROPORTION OF ADMIXTURE 1	0.65343	-0.00025	0.65343
PROPORTION OF ADMIXTURE 2	0.30931	-161084.98107	0.30931
PROPORTION OF ADMIXTURE 3	0.03726	-161232.32720	0.03757

NEGATIVE VARIANCES ARE PRINTED IN STD. ERROR COLUMN

JOINT LOG LIKELIHOOD OF THE SAMPLE = -0.1395082607D 02

MAXIMUM LAST ITERATION CHANGE IN ESTIMATES 0.0

CONVERGENCE IN 19 ITERATIONS

These results indicate that a mixture of <u>two</u> distributions is appropriate. The "negative variances" are due to fitting too many parameters, causing W to be close to singular. (See p. III-12).

RESULTS VERIFIED BY SEARCH AROUND FINAL ESTIMATES

NEWTON-RAPHSON APPLIED TO FINAL ESTIMATES GAVE LOWER LIKELIHOOD

VARIANCE-COVARIANCE MATRIX

0.004597	-0.002278	0.000585	0.001242	-0.000136	-24.120590	24.120726
-0.002278	0.021071	-0.034386	-0.002228	0.002320	-31.250745	31.248424
0.000585	-0.034386	0.458222	-0.002058	0.011067	-12.132232	12.121164
0.001242	-0.002228	-0.002058	0.000877	-0.000041	-1.429201	1.429242
-0.000136	0.002320	0.011067	-0.000041	-0.000250	-73.672940	73.673190
-24.120590	-31.250745	-12.132232	-1.429201	-73.672940	*************	161158.654011
24.120726	31.248424	12.121164	1.429242	73.673190	161158.654011	*************

ROUNDING ERRORS MAY BE AFFECTING THE FOLLOWING ELEMENTS
IN THE MATRIX OF SECOND PARTIAL DERIVATIVES.
ELEMENT NUMBERS DO NOT COUNT PARAMETERS WHICH ARE DEPENDENT OR HAVE CONVERGED TO A BOUND.

I	J	D1(I,J)	D2(I,J)	D3(I,J)
6	2	-0.00571987	-0.00573497	-0.00558587
6	3	-0.00093077	-0.00109796	-0.00095481
6	4	-0.01722947	-0.01676391	-0.01490116
6	5	0.01712900	0.04568874	0.27416667
6	6	-0.00000037	-0.00000040	-0.00000182

FIRST DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
WITH RESPECT TO INDEPENDENT PARAMETERS NOT FIXED

0.000084	-0.000134	0.000016	0.008367	0.000400	0.0

NEGATIVES OF THE SECOND DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
WITH RESPECT TO INDEPENDENT PARAMETERS NOT FIXED

350.269748	-9.934766	-1.196408	-529.594232	-98.884103	-0.000507
-9.934766	120.391951	12.467015	355.201730	-69.265580	0.005720
-1.196408	12.467015	3.537809	42.789725	-8.344373	0.000931
-529.594232	355.201730	42.789725	2917.400555	-78.594517	0.017229
-98.884103	-69.265580	-8.344373	-78.594517	102.107125	-0.017129
-0.000507	0.005720	0.000931	0.017229	-0.017129	0.000000

NO. OF LIKELIHOOD EVALUATIONS 623

ANY OUTPUT PROGRAMMED IN SECTION (VI) OF THE USER SUPPLIED LIKELIHOOD SUBROUTINE IS PRINTED HERE.

C. Example 3

The results of example 2 indicated that, when logarithms of the original data are taken, a mixture of only two normal distributions is appropriate. In example 3 logarithms of the original data are taken and a mixture of two normal distributions fitted. The data input deck is exactly the same as for example 2, except for the program parameter cards; these are:

```
RED CELL SODIUM DATA. SAMPLE 3 MALES.
0.001               0.5              50  5  5   1 111 102  1102        13  11
  5   1  4
7.336      11.509   4.688      0.653      0.347
0.0        0.0      0.001      0.0        0.0
100.       100.     100.       1.0        1.0
   47     2
```

THE REST OF THE DATA INPUT DECK IS THE SAME AS FOR EXAMPLES 1 AND 2.

Only the summary of the results is reproduced here; it should be noted that the estimates are virtually the same as those for example 2, but now there are no longer any "negative variances".

SUMMARY OF RESULTS

MAXIMUM LIKELIHOOD ESTIMATES FOR

MIXTURE OF UNIVARIATE NORMAL DISTRIB.

RED CELL SODIUM DATA. SAMPLE 3 MALES. EX. 3 IN WRITE UP
NO. OF DISTRIBUTICNS 2
TRANSFORMATION CODES 2 0.0 0.0

ML PROGRAM APR. 10, 1973 VERSION

DIRECT SEARCH OF LIKELIHOCD SURFACE

PARAMETER	LOWER BOUNDS	INITIAL ESTIMATES	UPPER BOUNDS
MEAN OF DISTRIBUTION 1	-10000.00000	1.99279	4.60517
MEAN OF DISTRIBUTION 2	-10000.00000	2.44313	4.60517
COMMON VARIANCE	0.00001	0.06076	1.29602
PROPORTION OF ADMIXTURE 1	0.0	0.65300	1.00000
PROPORTION OF ADMIXTURE 2	0.0	0.34700	1.00000

LIKELIHOOD OF INITIAL ESTIMATES -0.1395085263D 02

MAXIMUM RELATIVE ESTIMATE CHANGE REQUESTED 0.00100 USED 0.00100

INITIAL STEP SIZE 0.50000

THE FOLLOWING PARAMETERS ARE FUNCTIONALLY DEPENDENT
5

PARAMETER	FINAL ESTIMATES	STD. ERROR	APPROXIMATE VALUE IN ORIGINAL SCALE
MEAN OF DISTRIBUTION 1	1.99339	0.09007	7.34035
MEAN OF DISTRIBUTION 2	2.44325	0.14387	11.51038
COMMON VARIANCE	0.06085	0.02960	4.69695
PROPORTION OF ADMIXTURE 1	0.65344	0.18254	0.65344
PROPORTION OF ADMIXTURE 2	0.34656	0.18254	0.34656

JOINT LOG LIKELIHOOD OF THE SAMPLE = -0.1395082607D 02

MAXIMUM LAST ITERATION CHANGE IN ESTIMATES 0.0014277

CONVERGENCE IN 4 ITERATIONS

RESULTS VERIFIED BY SEARCH AROUND FINAL ESTIMATES

FINAL ESTIMATES IMPROVED BY ONE NEWTON-RAPHSON ITERATION

VARIANCE-COVARIANCE MATRIX IS BASED ON ESTIMATES OBTAINED BY DIRECT SEARCH WHICH ARE:

1.99280

2.44308

0.06076

0.65267

0.34799

AT END OF DIRECT SEARCH LIKELIHOOD WAS -0.1395084749D 02 . MAXIMUM PREVIOUS ITERATION CHANGE 0.0000001

VARIANCE-COVARIANCE MATRIX

0.008113	0.002463	0.001416	0.010793	-0.010793
0.002463	0.020698	-0.001933	0.016659	-0.016659
0.001416	-0.001933	0.000876	0.000565	-0.000565
0.010793	0.016659	0.000565	0.033320	-0.033320
-0.010793	-0.016659	-0.000565	-0.033320	0.033320

FIRST DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
WITH RESPECT TO INDEPENDENT PARAMETERS NOT FIXED

0.075798	-0.002069	-0.015468	0.000004

NEGATIVES OF THE SECOND DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
WITH RESPECT TO INDEPENDENT PARAMETERS NOT FIXED

350.452737	-11.536301	-528.333744	-98.794408
-11.536301	149.538976	398.770502	-77.789006
-528.333744	398.770502	2925.702691	-77.838937
-98.794408	-77.789006	-77.838937	102.225728

NO. OF LIKELIHOOD EVALUATIONS 199

SUMMARY OF RESULTS

MAXIMUM LIKELIHOOD ESTIMATES FOR

MIXTURE OF UNIVARIATE NORMAL DISTRIB.

RED CELL SODIUM DATA. SAMPLE 3 MALES. EX. 3 IN WRITE UP
NO. OF DISTRIBUTIONS 2
TRANSFORMATION CODES 2 0.0 0.0

ML PROGRAM APR. 10, 1973 VERSION

NEWTON-RAPHSON METHOD
WITHOUT RECALCULATING VARIANCE-COVARIANCE MATRIX

PARAMETER	LOWER BOUNDS	INITIAL ESTIMATES	UPPER BOUNDS
MEAN OF DISTRIBUTION 1	-10000.00000	1.99339	4.60517
MEAN OF DISTRIBUTION 2	-10000.00000	2.44325	4.60517
COMMON VARIANCE	0.00001	0.06085	1.29602
PROPORTION OF ADMIXTURE 1	0.0	0.65344	1.00000
PROPORTION OF ADMIXTURE 2	0.0	0.34656	1.00000

LIKELIHOOD OF INITIAL ESTIMATES -0.1395082607D 02

MAXIMUM RELATIVE ESTIMATE CHANGE REQUESTED 0.00100 USED 0.00100

THE FOLLOWING PARAMETERS ARE FUNCTIONALLY DEPENDENT
5

PARAMETER	FINAL ESTIMATES	STD. ERROR	APPROXIMATE VALUE IN ORIGINAL SCALE
MEAN OF DISTRIBUTION 1	1.99339	0.09060	7.34035
MEAN OF DISTRIBUTION 2	2.44325	0.14413	11.51038
COMMON VARIANCE	0.06085	0.02983	4.69695
PROPORTION OF ADMIXTURE 1	0.65344	0.18287	0.65344
PROPORTION OF ADMIXTURE 2	0.34656	0.18287	0.34656

JOINT LOG LIKELIHOOD OF THE SAMPLE = -0.1395082607D 02

MAXIMUM LAST ITERATION CHANGE IN ESTIMATES 0.0000172

CONVERGENCE IN 1 ITERATIONS

VARIANCE-COVARIANCE MATRIX

0.008209	0.002402	0.001456	0.010896	-0.010896
0.002402	0.020772	-0.001950	0.016613	-0.016613
0.001456	-0.001950	0.000890	0.000613	-0.000613
0.010896	0.016613	0.000613	0.033443	-0.033443
-0.010896	-0.016613	-0.000613	-0.033443	0.033443

FIRST DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
WITH RESPECT TO INDEPENDENT PARAMETERS NOT FIXED

0.000114	-0.000089	0.008693	0.00009[illegible]

NEGATIVES OF THE SECOND DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
WITH RESPECT TO INDEPENDENT PARAMETERS NOT FIXED

350.276063	-11.132142	-529.603062	-98.885418
-11.132142	148.861841	397.982546	-77.614981
-529.603062	397.982546	2917.434327	-78.622009
-98.885418	-77.614981	-78.622009	102.115902

NO. OF LIKELIHOOD EVALUATIONS 377

ANY OUTPUT PROGRAMMED IN SECTION (VI) OF THE USER SUPPLIED LIKELIHOOD SUBROUTINE IS PRINTED HERE.

VIII. RELATED READING

The following articles contain discussions of the problem of function maximization and give references to other methods.

Fletcher, R., Function minimization without evaluating derivatives -- a review, Comput. J., 8 (1965), pp. 33-41.

Powell, M. J. D., A survey of numerical methods for unconstrained optimization, SIAM Review, 12, No. 1 (1970), pp. 79-97.

IX. DISCLAIMER

Although this program has been extensively tested, no warranty, expressed or implied, is made as to the accuracy and functioning of the program. No responsibility is assumed by the authors. If any errors are found, please contact the Department of Biostatistics, School of Public Health, University of North Carolina, Chapel Hill, North Carolina, 27514.