NONMEM Users Guide -- Part II

Users Supplemental Guide

April 1988

by

Stuart L. Beal
And
Lewis B. Sheiner

Electronic copy produced
March 2008, August 2011

TABLE OF CONTENTS

A. Introduction

This document describes features of the NONMEM system that are not described in the Users Basic Guide, i.e., NONMEM Users Guide, Part I. Note moreover, that some features described here are not available with Version I, or with just Version I Level 1. Such circumstances wherein features are not available, are given in the text.

The computer output shown in the figures from the examples of section D is obtained from the Control Data Corporation version of NONMEM, rather than from the IBM version. The CDC version is not being distributed. The computer output shown in the figures from the example of section C is obtained from the IBM version of NONMEM, Version II Level 1.

We take this opportunity to mention some miscellaneous points that were not made clear in the Users Basic Guide.

> The same ID data item may be used to identify two different individual records, providing these records are not contiguous in the data set.
>
> When the MDV data item of a data record is 1, the DV data item of that record is not included in a SCATTERPLOT in which DV data items are plotted on either the ordinate or abscissa axis.
>
> The width of an A format code appearing in a FORMAT record should not exceed 4 (the no. of bytes in an IBM word). A data item read with such a code should not be tabled.

B. Constraints on Omega and Sigma

B.1 Introduction

The omega and sigma matrices are always constrained to be positive semidefinite, but they also may be constrained in certain special ways. This is the topic discussed in this section. Particular examples are given here without the aid of computer output. Some examples with computer output are given in sections C and D. Some general background is given first in section B.2, and then a discussion of the general use of the STRUCTURE and BLOCK SET records to implement constraints is given in section B.3.

B.2 Background

Consider either the omega or sigma matrix, and denote it by M. Recall that this matrix is the covariance matrix of either the η effects or the ε effects. One constraint on M that can be implemented is that M have the form

$$M = \begin{pmatrix} M_1 & & & 0 \\ & M_2 & & \\ & & \ddots & \\ 0 & & & M_n \end{pmatrix}$$

dimension of $M_i = d_i$ for all $1 \leq i \leq n$

where the d_i are fixed integers. That is, M is constrained to be in block diagonal form, where the submatrices along the diagonal have given dimensions. These submatrices are called the diagonal blocks of M. For example, suppose that there are three η effects and that η_3 is to be regarded as uncorrelated with η_1 and η_2 Then M should be constrained to be in block diagonal form with n = 2, and the dimensions of M_1 and M_2 should be 2 and 1 respectively. If η_2 is to be regarded as uncorrelated with η_1 and η_3, then simply renumber these random effects (in particular, permute $\eta_2$and η_3), and the desired end can be achieved by constraining M to be in the same block diagonal form as the one just given. If the three random effects are to be regarded as completely pairwise uncorrelated, i.e. all off diagonal elements of M are to be zero, then M should be constrained to be in block diagonal form with n = 3, and the dimensions of the three diagonal blocks should all be 1. Of course, the constraint that M be diagonal can be implemented more easily by using the DIAGONAL record as explained in the NONMEM Users Guide Part I. The block diagonal constraint is the basic constraint in terms of which the other constraints are expressed.

Another constraint on M that can be implemented involves the notion of a block set. A block set is a set of consecutively ordered diagonal blocks, once M is expressed in block diagonal form. For example, $\{M_3, M_4, M_5\}$, $\{M_5, M_6\}$, and $\{M_1\}$ are all block sets. However $\{M_1, M_3\}$ is not a block set since M_1 and M_3 are not consecutively ordered. Now the n diagonal blocks may be partitioned into block sets. The partition is called a block set partition. In each block set of the partition, the blocks of the set are constrained to be equal. Therefore, all blocks of a particular block set must have the same dimension. For example, in the above situation where there are three η effects and n = 3, the variances of the first two effects can be constrained to be the same by partitioning the blocks into block sets $\{M_1, M_2\}$ and $\{M_3\}$. The variances of all three effects can be constrained to be the same by forming the single block set $\{M_1, M_2, M_3\}$.

The last type of constraint on M that can be implemented also involves the notion of a block set partition. All the blocks of any particular block set can be constrained to equal some given matrix. For example, in the above situation where there are three η effects and n = 2, the block M_1 can be constrained to equal the two dimensional matrix A. To do this one must first partition the diagonal blocks into the block sets $\{M_1\}$ and $\{M_2\}$. To take another example, in the above situation where there are three η effects and n = 3, blocks M_1 and M_2 may be constrained to both equal the number a, while M_3 may be constrained to equal the number b. To do this the diagonal blocks should first be partitioned into the block sets $\{M_1, M_2\}$ and $\{M_3\}$. To take yet another example, if M is to be constrained to equal a matrix A, then only one block set need be formed, viz, $\{M\}$. If, though, A is diagonal, then one could alternatively take the partition $\{M_1\}$, $\{M_2\}$, etc. of block sets, each of which consists of a single one dimensional block, and specify that each block is to equal a diagonal element of A.

B.3 Implementation

The three types of constraints described in section B.2 are implemented by using the STRUCTURE and BLOCK SET records. Suppose one desires to impose one of these types of constraints on the omega matrix. A block set partition must be constructed. This is true even if there is only one block per block set (as is the case with the first type of constraint described in section B.2). The partition is specified in part with the STRUCTURE record for omega. This integer format record has two fields for each of the block sets of the block set partition. For each block set, the number of blocks in the set and the common dimension of all the blocks in the set are placed in the first and second of the fields, respectively, corresponding to this block set. The pairs of fields are ordered on the record as the block sets are ordered in the block set partition. For example, if there are two block sets such that the size of the first set and the dimension of its blocks are 2 and 1, respectively, and the size of the second set and the dimension of its blocks are 1 and 2, respectively, then the STRUCTURE record is as in Example B. 3.i.

Example B.3.i:

```
STRC          2    1    1    2

column no.:   1    1    2    2
              2    6    0    4
```

In addition, fields 6 and 7 of the initial STRUCTURE record must have appropriate values. A zero or blank is placed in field 6, and the number of block sets of the partition is placed in field 7. (Note that all examples in NONMEM Users Guide, Part I, involve omega matrices that are either constrained to be diagonal or are unconstrained. In the latter case a trivial constraint is actually being imposed where there is only one block set, and this block set has only one block. Therefore, throughout that document all discussion concerning the STRUCTURE record for omega indicates that there are just two fields of that record, in which are placed the values 1 and the dimension of omega. Also therefore, it is indicated that whenever a STRUCTURE record for omega appears, the values 0 and 1 are placed in fields 6 and 7, respectively, of the initial STRUCTURE record.)

The way in which the block set partition is implemented for the sigma matrix is similar to the way this is done for the omega matrix. A STRUCTURE record for sigma appears. In addition, a zero or blank is placed in field 8 of the initial STRUCTURE record, and the number of block sets of the partition is placed in field 9 of that record. The general formats of the initial STRUCTURE record and the STRUCTURE record for omega (and sigma) are given, respectively, in Tables B.3.i and B.3.ii, below.

Table B.3.i Initial STRUCTURE record format ***(See Introduction to NONMEM VI March 2008 for complete format.)***

Field no.	**Value**	**Function**
1	between 0 & 70	length of theta vector
2	between 0 & 70	length eta vector
3	between 0 & 70	length of epsilon vector
4	blank	
5	blank	
6	0 or blank 1	omega constrained with a block set partition omega constrained to be of simple diagonal form
7	0 or blank between 1 & 70	only if field 6 has value 1 number of block sets for omega
If the dimension of Σ is 0, the following fields may be ignored.		
8	0 or blank 1	sigma constrained with a block set partition sigma constrained to be of simple diagonal form
9	0 or blank between 1 & 70	only if field 8 has value 1 number of block sets for sigma

TABLE B.3.ii STRUCTURE record for omega (or sigma)

Field no.	**Value**	**Function**
1	between 1 & 70	size of 1st block set
2	between 1 & 70	Dimension of blocks in 1st block set
3	between 1 & 70	size of 2nd block set
4	between 1 & 70 etc.	Dimension of blocks in 2nd block set

The initial estimate for omega is specified on a contiguous series of BLOCK SET records for omega, one for each block set of the partition. The common initial estimate for the blocks of the ith block set is placed on the ith BLOCK SET record. There is a matrix, $\mathbf{\Omega}_i$, such that each block of the ith block set equals $\mathbf{\Omega}_i$, and the estimates of the individual elements of the upper triangular part of $\mathbf{\Omega}_i$ are placed in the successive fields of the ith BLOCK SET record according to the ordering: ω_{i11}, ω_{i12}, …, $\omega_{i1K(i),}$ ω_{i22}, ω_{i23},…,□ $\omega_{i2K(i)}$, …, $\omega_{iK(i)K(i),}$ where K(i) is the dimension of $\mathbf{\Omega}_i$ (Recall that $\mathbf{\Omega}_i$ is symmetric.) With the third type of constraint, $\mathbf{\Omega}_i$, in particular, may be constrained to equal some given matrix, $A_{i.}$ In this case. A_i should be taken to be the initial estimate of $\mathbf{\Omega}_{i.}$ Moreover, a special character should be placed on the ith BLOCK SET record, namely a 1 in position 8. The initial estimate for sigma is specified on contiguous series of BLOCK SET records for sigma in exactly the manner the way the initial estimate for omega is specified. The BLOCK SET records for sigma follow those for omega within the Model Specification records. Of course, if on the initial STRUCTURE record, a 1 is placed in either field 6 or 8, then the BLOCK SETS for omega or sigma, respectively, are replaced by a single DIAGONAL record. The general format for the BLOCK SET records is given in Table B.3.iii.

Table B.3.iii BLOCK SET record format

Field no.	**Value**	**Function**
1		initial estimate of (1,1) element of block
2	etc.	initial estimate of (1,2) element of block
In addition, a 1 is placed in position 8 if this block is constrained to equal the initial estimate.		

C. Simple Bayesian Nonlinear Regression

C.1 Introduction

An example of simple nonlinear regression is discussed in section C of NONMEM Users Guide, Part I. This example is continued in this section in order to illustrate the discussion concerning constraints given in section B and in order to illustrate several additional NONMEM features. The first additional feature to be illustrated is the Theta Constraint Option, by which one may exercise some control over the way the constraints on the theta elements (see section C.4 of NONMEM Users Guide, Part I) are implemented inside the NONMEM program itself. The second additional feature gives the user the option to add a forty-five degree line onto a scatterplot. This can be useful when, for example, in a simple nonlinear regression, predictions are plotted against observations. The last additional feature gives the user the option to modify the way in which the objective function is computed (while the statistical model itself remains fixed); this involves use of a user-supplied subroutine, CRIT. The basic computation that is presented here, one of a simple Bayesian nonlinear regression, does not, however, depend on the presence of any of the additional features just mentioned, and it may be carried out as illustrated below with only NONMEM features described in NONMEM users Guide, Part I, and constraints on omega as discussed in section B of this document.

In section C.2 the theoretical aspects of the example are discussed. In section C.3, implementation of the example is discussed. The three additional features that are mentioned above are discussed in sections C.4-C.6. The reader can go immediately to these sections if he wishes only to review the use of these additional features.

C.2 An Example

We continue the example of section C in the NONMEM Users Guide, Part I. First, we reparametrize the function, f, as follows

$$f(\theta_1,\theta_2,\theta_3,X_1,X_2)=\frac{\theta_1\theta_2X_1}{\theta_3(\theta_1-\theta_2)}(\exp(-\theta_2X_2)-\exp(-\theta_1X_2))$$

In words, we have redefined θ_3 to be the product of θ_2 and the original θ_3. In pharmacokinetic language, θ_3 now denotes drug clearance, where before it denoted volume.

The data in the old example are ten plasma concentrations from a single subject, S, from a certain population. The (true) value of $\theta=(\theta_1,\theta_2,\theta_3)$ varies from subject to subject within this population. Although in this new example we are again interested in the (true) value of θ for S, here we shall consider an estimate of this value that is based on knowledge of how θ varies in the population, in addition to the plasma concentration data from S. The population itself has been studied by measuring plasma concentrations from each of a sample of twelve subjects chosen from the population. This larger data set was used in the example in section F of the

NONMEM Users Guide, Part I, wherein estimates, $\bar{\theta} = (\bar{\theta}_1, \bar{\theta}_2, \bar{\theta}_3)$ and C, of the mean and covariance, respectively, of θ in the population were obtained. Actually, $\bar{\theta}_3$ was obtained for each of a family of subpopulations obtained by "stratifying" the population by body weight. The estimate, $\bar{\theta}_3$ is characterized then as a function of weight. The relationship of $\bar{\theta}_3$ to weight is a simple proportional one. Since the weight of S, in particular, is 72.4, $\bar{\theta}_3$ for the particular subpopulation of interest is given by the proportionality constant, .0363, times 72.4, i.e. 2.63. On the other hand, in the analysis of the population data, θ was assumed to have homogeneous covariance in the population. Another important population parameter was estimated in the example of section F of NONMEM Users Guide, Part I. This parameter (denoted in that example by Σ) is the variance of the plasma concentrations from any population member, given that member's value of θ, and it is independent of θ. Denote its estimate by σ^2 .

Consequently, in addition to the ten measurements from S there are three other measurements from the population _that have information about θ for S, namely: $\bar{\theta}_1, \bar{\theta}_2$,and $\bar{\theta}_3$. More precisely,

$$\bar{\theta}_i = \theta_i + \eta_{io} \quad i = 1,2,3$$

where

$$\mathrm{E}(\eta_{10}, \eta_{20}, \eta_{30}) = 0$$
$$Cov(\eta_{10}, \eta_{20}, \eta_{30}) = C.$$

(Of course, C is only an estimate of the covariance, and the assumption that the expectation is 0 is compromised by the fact that the $\bar{\theta}_i$ are only estimates of the true means of the θ_i. However, some people feel that often one is better off using some prior (population) information about the unknown parameter one is trying to estimate, rather than none at all, even if this information is not very precise.) Combining this model with the model for the ten plasma concentrations, we obtain

$$\begin{aligned}
&Y_{io} = \theta_i + \eta_{io} && i = 1,2,3 \\
&Y_{1j} = f(\theta_1, \theta_2, \theta_3, X_{1j}, X_{2j}) + \eta_{4j} && j = 1,2,3,\ldots,10 \\
&E = (\eta_{1j}, \eta_{2j}, \eta_{3j}, \eta_{4j}) = 0 && j = 0,1,2,\ldots,10 \\
&Cov = (\eta_{1j}, \eta_{2j}, \eta_{3j}, \eta_{4j}) = \begin{pmatrix} C & 0 \\ 0 & \sigma^2 \end{pmatrix} && j = 0,1,2,\ldots,10
\end{aligned}$$

Here, Y_{io}= $\bar{\theta}$ and the Y_{1j} are the ten plasma concentrations. This is an example of a model with equivalently nested random effects and NONMEM may be used. (See section D.6 of NONMEM Users Guide, Part I.) Of course, one must be able to constrain the covariance matrix of the η_{ki} as specified above. This, though, is the major point of this example, and the set up for NONMEM is described in the next section.

The (default) objective function for this particular model is

$$O(\theta_1,\theta_2,\theta_3)=Y_0C^{-1}Y_0^T+\sum_{j=1}^{10}(Y_{1j}-f(\theta_1,\theta_2,\theta_{3,}X_{1j},X_{2j}))^2/\sigma^2$$

plus another (log) term that is independent of θ**,** as may be seen by regarding the objective function for the general model with equivalently nested random effects. Here $Y_0=(Y_{10},Y_{20},Y_{30})$. Note that σ^2 is not an argument of this function, in contrast with the situation in the old example, since in this new example we assume it to be a known constant. Instead of viewing the population information as giving rise to a model with equivalently nested random effects, one may take another view that leads to the same objective function. Assume that θ is normally distributed in the population with mean $\bar{\theta}$ and covariance C. This multivariate distribution is taken as a prior distribution on θ for S. Assume also that the conditional distribution of Y_{1j}, given θ, is normal with mean $f(\theta_1,\theta_2,\theta_3,X_{1j},X_{2j})$ and variance σ^2. Then using Bayes formula (1, page 334) the negative log of the posterior density of θ, may be seen to equal the above objective function, again up to a term independent of θ. From this point of view the estimate of θ obtained from minimizing this objective function is the mode of the posterior distribution.

C.3 Implementation of the Example

A code for PRED is given in Figure 1. There it may be seen that an additional data item is used to determine whether the DV data item is an observed concentration or an element of $\bar{\theta}$. If the value of this data item is zero, the DV data item is an observed concentration: if it is i, the DV data item is $\bar{\theta}_i$.

The problem specification is given in Figure 2. Upon examining the data, one sees that a fifth data item also has been added. This is the ID data item. It is there because the η_k are random individual effects.

In this example the block set partition for omega has two block sets, each of these block sets has a single block, and the two blocks have dimension 3 and 1. This is reflected in fields 6 and 7 of the initial STRUCTURE record and in the STRUCTURE record for omega. Note also that there is a one in position 8 of both BLOCK SET records, thus constraining the two blocks to C and σ^2, respectively, the values of which are specified on these records. There is no sigma matrix.

There is a two in field 5 of the initial STRUCTURE record, and there is a one in field 6 of the last SCATTERPLOT record. These specifications are discussed in sections C.4 and C.5.

Some output is shown in Figures 3-7. The summary of the problem specification is given in Figures 3 and 4. Note that there the block diagonal form of omega is schematically displayed as a patterned lower triangular matrix, and that the initial estimate of omega is displayed by giving the initial estimate of each of the two blocks separately. There is also an indication there that each block is constrained, or fixed, to its initial estimate. The final estimate is shown in Figure 5. The final estimate of omega is, of course, equal to the initial estimate. Some of the scatterplots are displayed in Figures 6 and 7. These are discussed in section C.5.

C.4 The Theta Constraint Option (Obsolete)

An element, θ_m, of theta may be bounded above and below by different finite numbers, a and b, respectively. This is accomplished by i) taking a function, T, defined on all real numbers, and whose range is the interval (b,a), ii) defining a new unconstrained parameter, φ_m satisfying

$$\theta_m = T(\varphi_m)$$

and iii) reparametrizing the functions f, g_{kij} and h_{lij} in terms of φ_m. All the user need do for these steps to be carried out is to use fields 4 and 5 of the initial STRUCTURE record and the LOWER and UPPER records as described in section C of NONMEM Users Guide, Part I. The value one placed in both fields 4 and 5 of the initial STRUCTURE record satisfies the requirement for that record, and all relevant examples in NONMEM Users Guide, Part I, show the value one in these fields. In this case T is given by

$$T(x) = b + (a-b)\sin^2(x).$$

A message that the arcsin transform is used appears on the summary of the problem specification: the arcsin is, of course the “inverse" of the sin function. The Theta Constraint Option allows the user to choose an alternative form for T. If the value two is placed in field 5 (see Figure 2), rather than one, T is given by

$$T(x) = b + (a-b)(\frac{\exp(x)}{1+\exp(x)}).$$

A message that the logit transform is used appears on the summary of the problem specification; see Figure 3. The logit is, of course, the inverse of the function:

$$x \rightarrow \frac{\exp(x)}{1+\exp(x)}.$$

Note that whatever choice for T is made, it applies to all elements of theta bounded above and below by different finite numbers. The complete format of the initial STRUCTURE record is given in Table C.4.i.

Table C.4.i. Initial STRUCTURE record format ***(See Introduction to NONMEM VI March 2008 for current format.)***

Field no.	**Value**	**Function**
1	between 0 & 70	length of theta vector
2	between 0 & 70	length eta vector
3	between 0 & 70	length of epsilon vector
4	blank	
5	blank	
6	0 or blank 1	omega constrained with a block set partition omega constrained to be of simple diagonal form
7	0 or blank between 1 & 70	only if field 6 has value 1 number of block sets for omega
If the length of the epsilon vector is 0, the following fields may be ignored.		
8	0 or blank 1	sigma constrained with a block set partition sigma constrained to be of simple diagonal form
9	0 or blank between 1 & 70	only if field 8 has value 1 number of block sets for sigma

C.5 The Unit Slope Line

The following discussion does not pertain to NONMEM Version I. Users of subsequent versions will notice that when one of the two data items defining the points on a scatterplot is the residual or weighted residual data item, a line, defining the zero residual for all values of the other data item, appears on the scatterplot (see Figure 6). The user may specify that another type of line appear on each scatterplot of any given family. This is the unit slope line, i.e. the line through the point (0,0) with slope equal to one. This line is useful, for example, when plotting the prediction data item against the DV data item (see Figure 7). If for some given family of scatterplots, the unit slope line is desired on each scatterplot of the family, the value one should be placed in field 6 of the SCATTERPLOT record defining the family. The complete format for the individual SCATTERPLOT record is given in Table C. 5.1.

Table C.5.i Individual SCATTERPLOT record format ***(See Introduction to NONMEM VI March 2008 for complete format.)***

Field no.	**Value**	**Function**
1	between 1 & 23	index of data items plotted or abscissa axis
2	between 1 & 23	index of data items plotted on ordinate axis
3	0 or blank 1 2	single member scatterplot a one-way partitioned scatterplot a two-way partitioned scatterplot
If value of field 3 is 0 or blank, the next two fields should be ignored.		
4	between 1 & 23	index of (1st) separator
If value of field 3 is 1, the next field should be ignored.		
5	between 1 & 23	index of (2nd) separator
6	0 or blank 1	no unit slope line appears unit slope line appears

C.6 CRIT

The reader will recall that the default objective function used in NONMEM is the sum of contributions from each individual and that the second term in the contribution from the jth individual is the sum of squared weighted residuals for the jth individual. The user may substitute for this term another function of the weighted residuals. For example, the term $\left|R_{j1}\right| + \left|R_{j2}\right| + \ldots \left|R_j n_j\right|$ may be used, where the R_{ji} are the n_j weighted residuals for the jth individual. This substitution is accomplished by including along with PRED another user-supplied subroutine, CRIT. A vector of weighted residuals for some individual is passed to CRIT, and CRIT should return the value of the function to be used on the weighted residuals. In fact, the function itself may vary from individual to individual since the number of the individual (in individual record ordering) is also passed to CRIT. Since the value of n_j varies with individual, this value is also passed. Finally, there is opportunity for initialization of CRIT since a last argument is passed to CRIT that functions such as the argument ICALL functions in PRED. The entire argument list is summarized in Table C.6.i.

Table C.6.i Arguments of CRIT

Argument	**Value**	**Function**
ICALL	0 1 2	first call to CRIT in the run first call to CRIT in current problem computation of function value required
J	positive integer	number of individual
N		number of observations for the individual
WRES		vector of weighted residuals
V		value of function

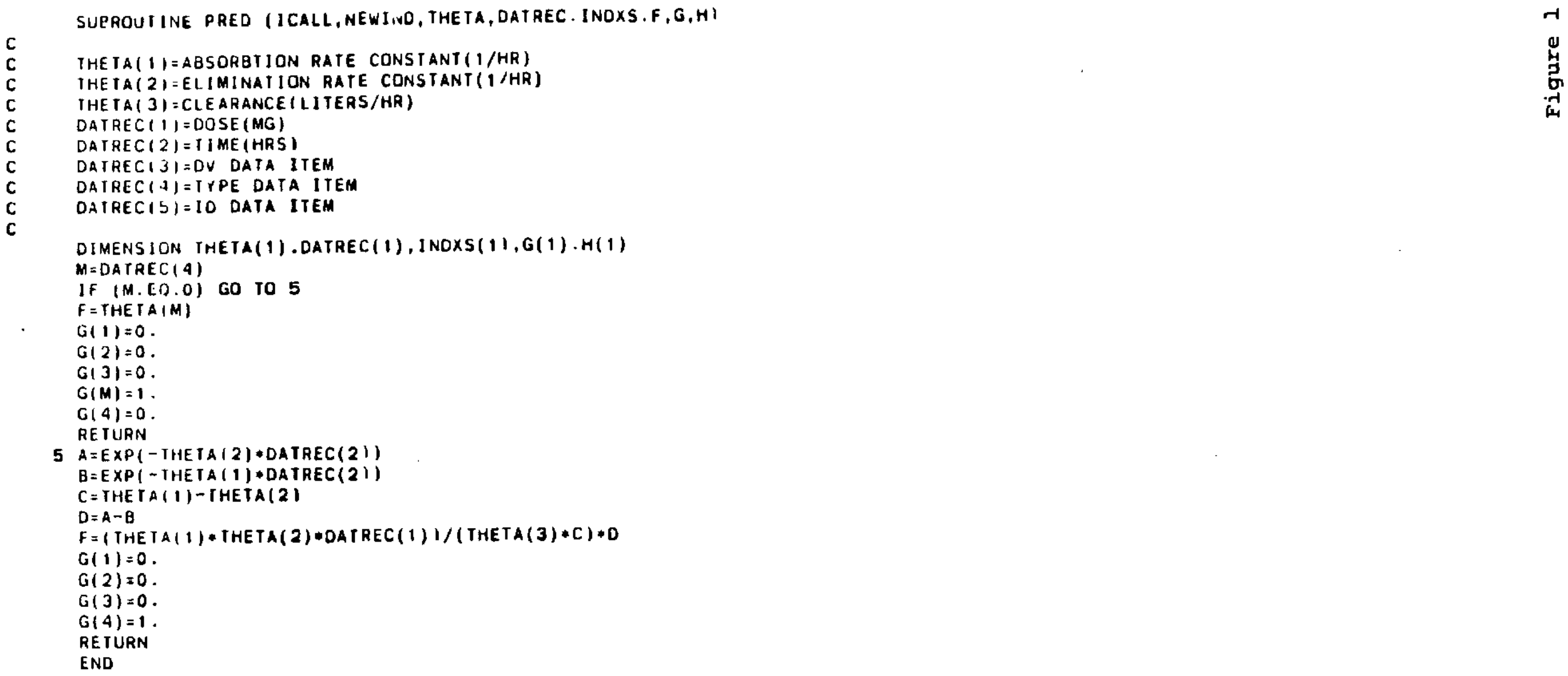
CRIT may be as complicated as seems necessary and appropriate. In particular, it may call other user-written subroutines to accomplish various parts of its task. The following DIMENSION statement should be included.

```
DIMENSION WRES (1)
```

In Figure 8 a code for CRIT used in the example is given. It may be easily seen that the function used is identical to the one used by default. Of course, CRIT is not needed when this particular function is used (and actually NONMEM does not compute a sum of squares when CRIT is not supplied, but rather NONMEM computes something which is equivalent, based on the particular form of the R_j.)

There is a subroutine named CRIT that is a part of the NONMEM package, and it has the same name as the subroutine described in this section. It is used automatically by NONMEM, as a result of employing the IBM Linkage Editor as described in NONMEM Users Guide, Part III, when a user-supplied CRIT is not present. However, it does not resemble the subroutine shown in Figure 8.)

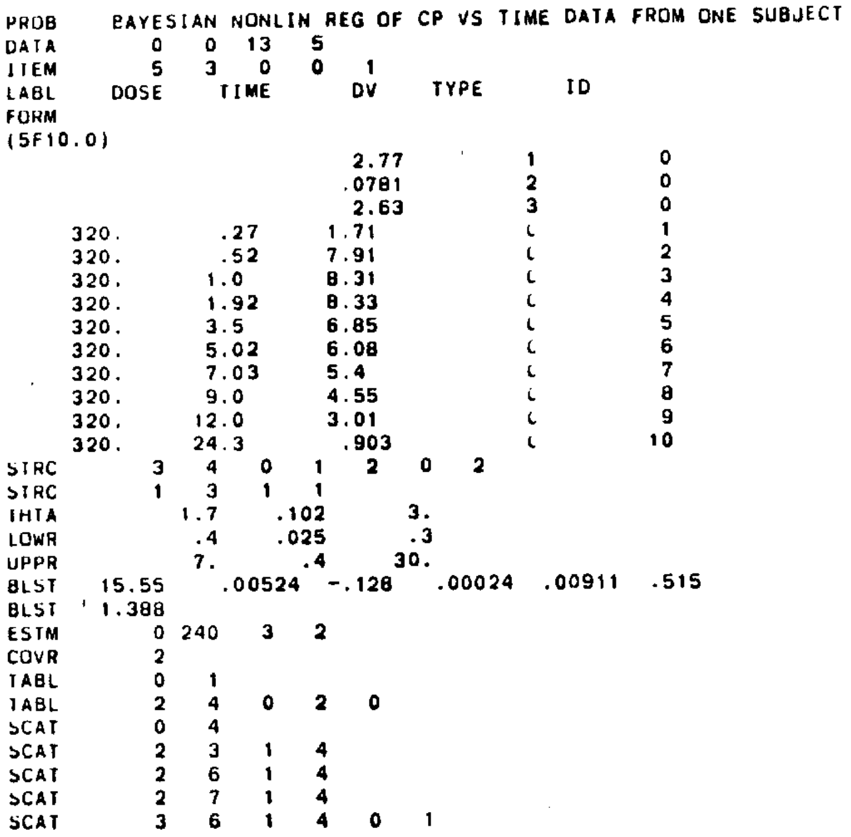
Figure 1



```
      SUPROUTINE PRED (ICALL,NEWIND,THETA,DATREC.INDXS.F,G,H)
C
C     THETA(1)=ABSORBTION RATE CONSTANT(1/HR)
C     THETA(2)=ELIMINATION RATE CONSTANT(1/HR)
C     THETA(3)=CLEARANCE(LITERS/HR)
C     DATREC(1)=DOSE(MG)
C     DATREC(2)=TIME(HRS)
C     DATREC(3)=DV DATA ITEM
C     DATREC(4)=TYPE DATA ITEM
C     DATREC(5)=ID DATA ITEM
C
      DIMENSION THETA(1).DATREC(1),INDXS(1),G(1).H(1)
      M=DATREC(4)
      IF (M.EQ.0) GO TO 5
      F=THETA(M)
      G(1)=0.
      G(2)=0.
      G(3)=0.
      G(M)=1.
      G(4)=0.
      RETURN
    5 A=EXP(-THETA(2)*DATREC(2))
      B=EXP(-THETA(1)*DATREC(2))
      C=THETA(1)-THETA(2)
      D=A-B
      F=(THETA(1)*THETA(2)*DATREC(1))/(THETA(3)*C)*D
      G(1)=0.
      G(2)=0.
      G(3)=0.
      G(4)=1.
      RETURN
      END
```

Figure 1

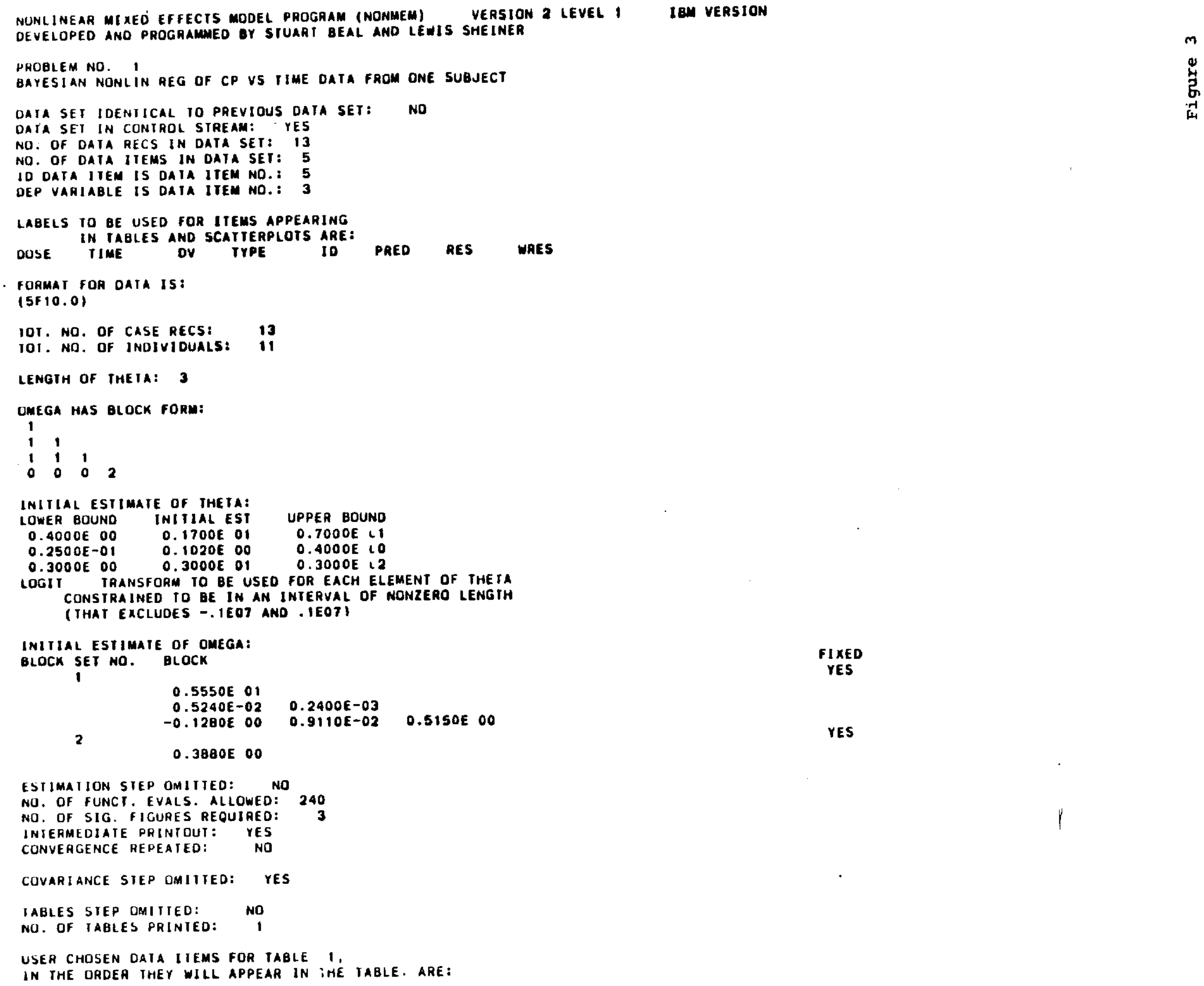
Figure 2



```
PROB     BAYESIAN NONLIN REG OF CP VS TIME DATA FROM ONE SUBJECT
DATA        0    0   13    5
ITEM        5    3    0    0    1
LABL     DOSE     TIME        DV     TYPE       ID
FORM
(5F10.0)
                               2.77            1          0
                              .0781            2          0
                               2.63            3          0
      320.        .27      1.71                0          1
      320.        .52      7.91                0          2
      320.       1.0       8.31                0          3
      320.       1.92      8.33                0          4
      320.       3.5       6.85                0          5
      320.       5.02      6.08                0          6
      320.       7.03      5.4                 0          7
      320.       9.0       4.55                0          8
      320.      12.0       3.01                0          9
      320.      24.3        .903               0         10
STRC        3    4    0    1    2    0    2
STRC        1    3    1    1
THTA          1.7     .102      3.
LOWR           .4     .025       .3
UPPR          7.        .4     30.
BLST   15.55     .00524  -.128   .00024  .00911  .515
BLST   1.388
ESTM        0  240    3    2
COVR        2
TABL        0    1
TABL        2    4    0    2    0
SCAT        0    4
SCAT        2    3    1    4
SCAT        2    6    1    4
SCAT        2    7    1    4
SCAT        3    6    1    4    0    1
```

Figure 2

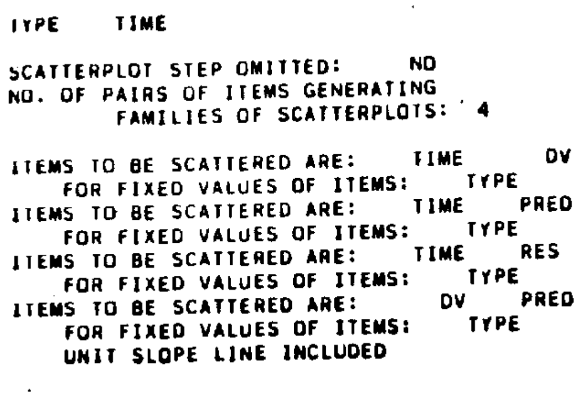
Figure 3



```
NONLINEAR MIXED EFFECTS MODEL PROGRAM (NONMEM)     VERSION 2 LEVEL 1     IBM VERSION
DEVELOPED AND PROGRAMMED BY STUART BEAL AND LEWIS SHEINER

PROBLEM NO.   1
BAYESIAN NONLIN REG OF CP VS TIME DATA FROM ONE SUBJECT

DATA SET IDENTICAL TO PREVIOUS DATA SET:    NO
DATA SET IN CONTROL STREAM:   YES
NO. OF DATA RECS IN DATA SET:  13
NO. OF DATA ITEMS IN DATA SET:  5
ID DATA ITEM IS DATA ITEM NO.:  5
DEP VARIABLE IS DATA ITEM NO.:  3

LABELS TO BE USED FOR ITEMS APPEARING
       IN TABLES AND SCATTERPLOTS ARE:
DOSE     TIME      DV     TYPE      ID     PRED     RES      WRES

FORMAT FOR DATA IS:
(5F10.0)

TOT. NO. OF CASE RECS:      13
TOT. NO. OF INDIVIDUALS:    11

LENGTH OF THETA:  3

OMEGA HAS BLOCK FORM:
 1
 1  1
 1  1  1
 0  0  0  2

INITIAL ESTIMATE OF THETA:
LOWER BOUND     INITIAL EST     UPPER BOUND
 0.4000E 00      0.1700E 01      0.7000E 01
 0.2500E-01      0.1020E 00      0.4000E 00
 0.3000E 00      0.3000E 01      0.3000E 02
LOGIT     TRANSFORM TO BE USED FOR EACH ELEMENT OF THETA
     CONSTRAINED TO BE IN AN INTERVAL OF NONZERO LENGTH
     (THAT EXCLUDES -.1E07 AND .1E07)

INITIAL ESTIMATE OF OMEGA:
BLOCK SET NO.   BLOCK                                                  FIXED
      1                                                                 YES
                 0.5550E 01
                 0.5240E-02   0.2400E-03
                -0.1280E 00   0.9110E-02   0.5150E 00
      2                                                                 YES
                 0.3880E 00

ESTIMATION STEP OMITTED:    NO
NO. OF FUNCT. EVALS. ALLOWED:  240
NO. OF SIG. FIGURES REQUIRED:    3
INTERMEDIATE PRINTOUT:   YES
CONVERGENCE REPEATED:     NO

COVARIANCE STEP OMITTED:   YES

TABLES STEP OMITTED:      NO
NO. OF TABLES PRINTED:     1

USER CHOSEN DATA ITEMS FOR TABLE  1,
IN THE ORDER THEY WILL APPEAR IN THE TABLE, ARE:
```

Figure 3

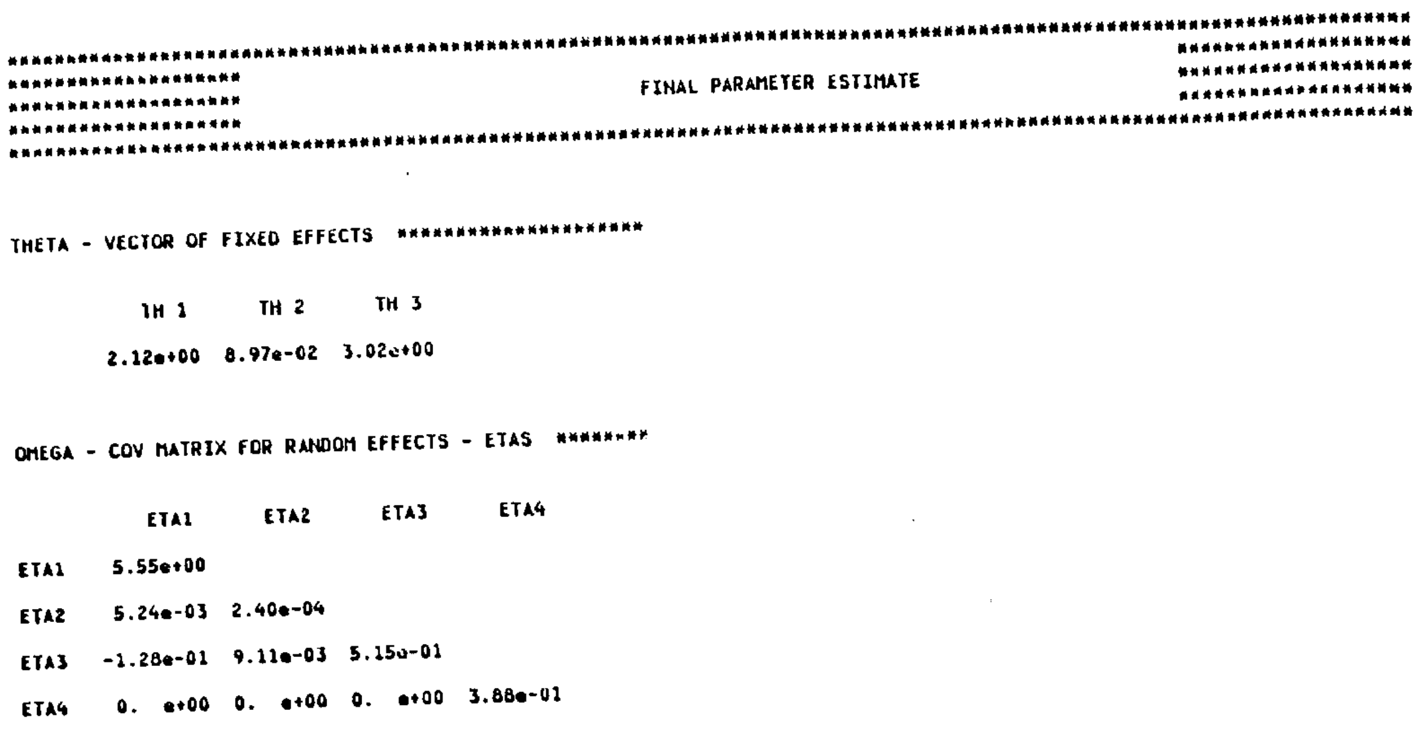
Figure 4



```
TYPE     TIME

SCATTERPLOT STEP OMITTED:      NO
NO. OF PAIRS OF ITEMS GENERATING
        FAMILIES OF SCATTERPLOTS:  4

ITEMS TO BE SCATTERED ARE:     TIME      DV
    FOR FIXED VALUES OF ITEMS:     TYPE
ITEMS TO BE SCATTERED ARE:     TIME    PRED
    FOR FIXED VALUES OF ITEMS:     TYPE
ITEMS TO BE SCATTERED ARE:     TIME    RES
    FOR FIXED VALUES OF ITEMS:     TYPE
ITEMS TO BE SCATTERED ARE:       DV    PRED
    FOR FIXED VALUES OF ITEMS:     TYPE
    UNIT SLOPE LINE INCLUDED
```

Figure 4

Figure 5



```
************************************************************************************************************************
********************                                                                              ********************
********************                              FINAL PARAMETER ESTIMATE                        ********************
********************                                                                              ********************
************************************************************************************************************************


THETA - VECTOR OF FIXED EFFECTS  *********************

            TH 1      TH 2      TH 3

         2.12e+00  8.97e-02  3.02e+00


OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

            ETA1      ETA2      ETA3      ETA4

ETA1     5.55e+00

ETA2     5.24e-03  2.40e-04

ETA3    -1.28e-01  9.11e-03  5.15e-01

ETA4     0.   e+00  0.   e+00  0.   e+00  3.88e-01
```

Figure 5

Figure 6

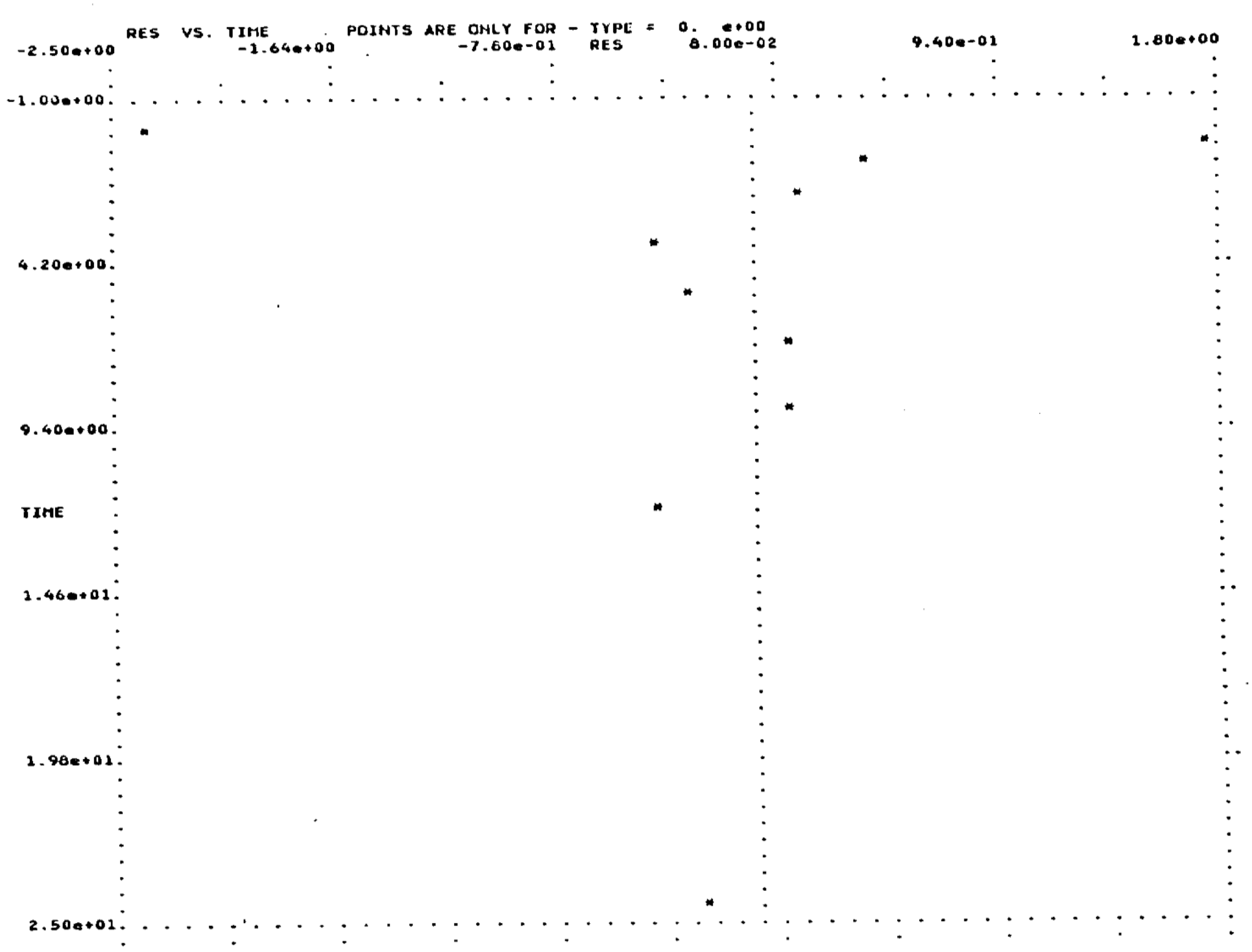


Figure 6

Figure 7

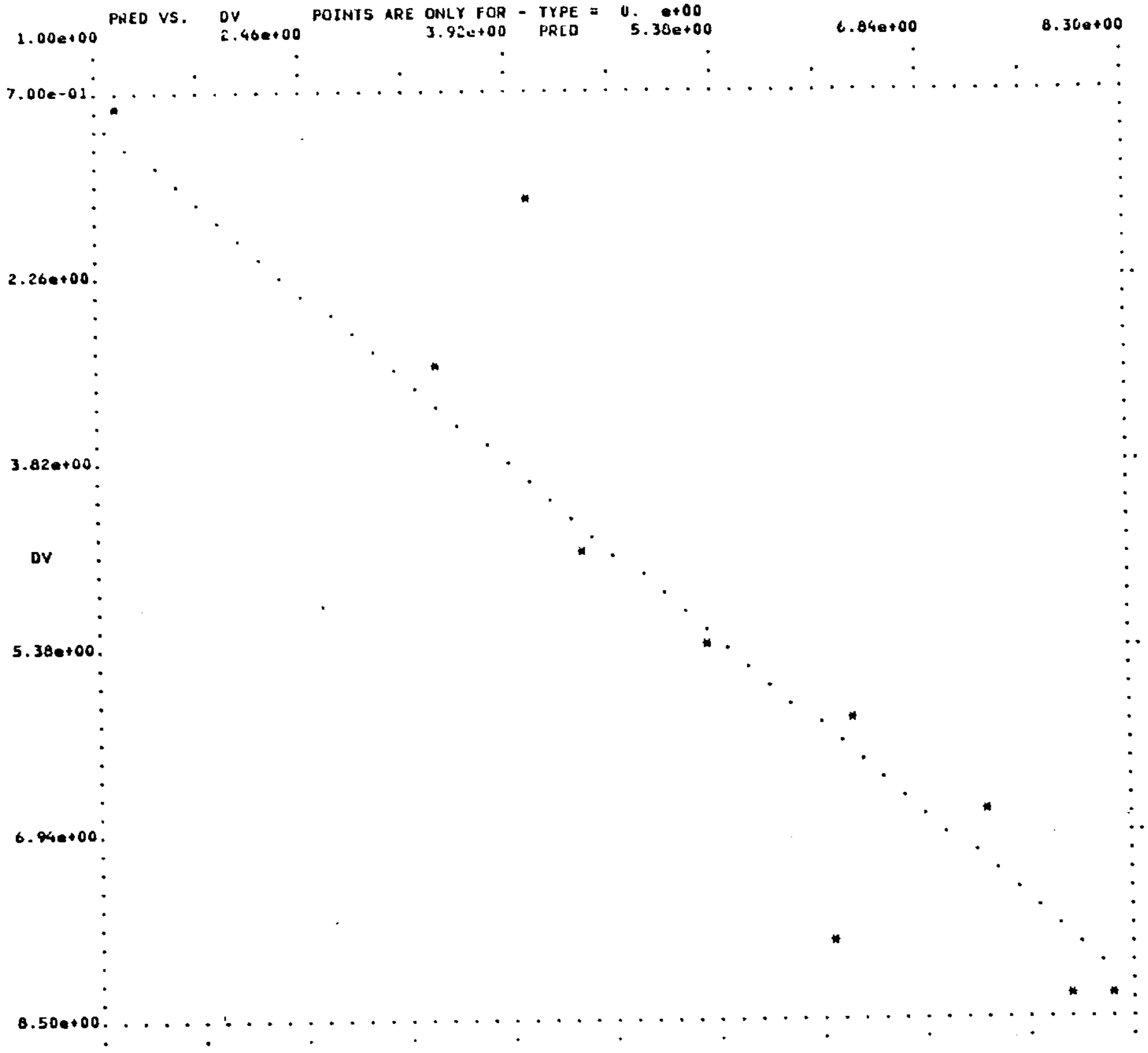
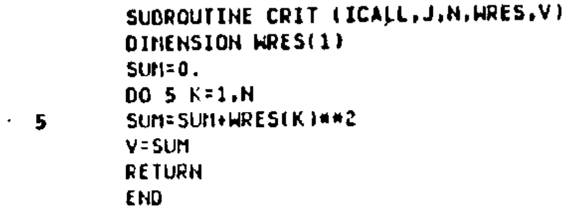


Figure 7

Figure 8



```
      SUBROUTINE CRIT (ICALL,J,N,WRES,V)
      DIMENSION WRES(1)
      SUM=0.
      DO 5 K=1,N
    5 SUM=SUM+WRES(K)**2
      V=SUM
      RETURN
      END
```

Figure 8

D. Linear Regression with One-Level Nested Random Effects

D.1 Introduction

An example of linear regression with one-level nested random effects is discussed in section E of NONMEM Users Guide, Part I (example 2). This example is continued in this section with three different variations, discussed in turn in sections D.2-D.4.

D.2 Example Involving Transgeneration of Data

D.2.1 Introduction

In the old example there occur pairs of DV data items, each pair consisting of a measured clearance and a measured elimination rate constant. From any pair, a derived measurement may be computed, the volume, which is the ratio of the clearance to the rate constant. In this new example the pairs of clearance and volume are available, rather than the pairs of clearance and rate constant. The data set looks the same except that when the type data item is 0, the DV data item is a clearance, and when the type data item is 1, the DV is a volume. The model to which the data will be fit is, however, the exact same one used in the old example. So, somehow the measured rate constants must be made available.

The rate constant could be computed in PRED. The idea here is based on the fact that when computing the value of the objective function, NONMEM takes as the value of the DV data item in a given data record, the value found in DATREC immediately after PRED has processed this data record and returned control to NONMEM. In our particular example then, the DV data item would be stored in A (some variable defined in PRED) when the type data item is 0, and the ratio of A to the DV data item would be computed and would replace the DV data item when the type data item is 1. However, a rate constant should only need be computed once, rather than with each pass through the data. Also, when the rate constant is returned from PRED as the DV data item, NONMEM does not store it in the data set or anywhere else where it could be available for tabling or scatterplotting. The returned value of the DV data item is used only in the computation of the objective function. Therefore, instead of computing the rate constant in PRED, the transgeneration feature is used. This is described in section D.2.2.

Four additional NONMEM features are described in sections

D.2.3-D.2.6. The first feature gives the user the option of not specifying the total number of data records in the data set on the DATA record. The second feature gives the user the option of specifying his own labels for the prediction, residual and weighted residual data items. The third feature gives the user several more options for computing the covariance matrix and obtaining intermediate printout involved in that computation. The fourth feature gives the user the option to obtain the eigenvalues of the correlation matrix.

D.2.2 Transgeneration (PASS)

The transgeneration of data is accomplished with a fair degree of generality and ease with NONMEM. Essentially, when ICALL, the first argument of PRED, is 0 or 1, PRED has access to the data set. Therefore, at run or problem initialization time, data items may be replaced with new ones computed in PRED. If a data item is input as a blank, or as any nonmeaningful number, replacing it with a new data item in PRED at initialization time, is tantamount to constructing an additional data item, although the actual number of data items per data record is unchanged. Moreover, when ICALL is 3, PRED also has access to the data set. Therefore, at problem finalization time data items that depend on the final estimate of theta may be computed and stored for subsequent tabling or scatterplotting.

The mechanism by which the data set is made available to PRED involves a NONMEM utility subroutine that PRED may call when ICALL is 0, 1 or 3. It is called PASS, and it has one integer argument, MODE. To pass through the data set, first call PASS with MODE set to 0, in order to initialize PASS. Then call it repeatedly, each time with MODE set equal to 2. With each such call another data record is passed to PRED and is stored in the DATREC argument of PRED. The first call produces the first data record, the second call produces the second data record, etc. The user need not keep counting the data records in order not to call PASS too many times. After each call PASS returns control to PRED with the value of MODE unaltered, except when there are no more data records to be passed, in which case MODE is reset by PASS to 0. Therefore, if after each call to PASS, PRED checks the value of MODE, PRED will know when one complete pass through the data records has occurred. PRED may initiate as many passes through the data records as is deemed necessary.

Before describing other points about PASS, let's look at an illustration. The code for PRED for the example is given in Figure 9. When ICALL is 1, PASS is called first with its argument, MODE, set equal to zero. PASS is thus initialized. Thereafter, it is called repeatedly at statement 15 with MODE equal to 2, and after each such call, the DV data item in DATREC(3) is replaced by the rate constant whenever the type data item is 1 (see discussion about the variable A at the beginning of section D.2.1). After each such call, the value of MODE is tested, and when it is 0, there is no more data to be transgenerated, and PRED returns control to NONMEM. Upon all subsequent calls to PRED, the rate constants, rather than the volumes, will be found in DATREC(3).

In addition to this transgeneration, when PRED is finalized, the predicted clearance and predicted volume are computed and stored in a new fifth data item. Recall that clearance is predicted by a linear function of weight. The constants of these linear functions depend on theta, whose final estimate is available in THETA. The predicted clearance or the predicted volume is stored in DATREC(5) according to whether the type data item is 0 or 1 respectively. To gain access to DATREC(5) and the type data item, PASS is used.

The problem specification for this example is given in Figure 10. Note that the number of data items per data record specified on the DATA control record is now five, and that the FORMAT record also reflects the presence of five data items per data record. Finally, note that blanks are input as the values of the fifth data item. It is immaterial what values are input since the values of the fifth data item are really obtained in PRED by transgeneration of the values of other data items.

The output from NONMEM from this example is, of course, essentially just that of the old example since only the data set before problem initialization is different, but at problem initialization the data set is transgenerated to look like the old data set. Scatterplots of observed clearance versus predicted clearance and of observed volume versus predicted volume were added to the output in this example and are given in Figures 11 and 12. Note the inclusion of the unit slope line (see section C.5)

There are two other values that may be assumed by MODE. If the user only wants to access the data records without modifying them, the value 1 may be used instead of 2. Use of the value 1 will safeguard inadvertent destruction of material in the data set, and it will also save some computing cost. Whatever value is used, 1 or 2, this value must be used on every call to PASS until PASS is reinitialized, with the following exception. If the value 1 is used, then at any call to PASS (after the initializing call) the value 3 may be used, in which case PRED tells PASS that the pass through the data records is to be terminated. A subsequent call to PASS must be an initializing call with MODE set equal to 0.

There are two ways a pass through the data records is terminated. Either the last data record of the data set has already been passed, and PASS returns with MODE equal to 0, or PRED sets MODE equal to 3 and calls PASS. In either case the first data record of the data set is again found in DATREC when PASS returns final control to PRED.

Lastly, during a pass through the data records, the value of the argument NEWIND in PRED changes as it would during a pass through the data records when ICALL is 2.

The possible values of MODE are summarized in Table D.2.2.i.

Table D.2.2.i Arguments of PASS

Argument	**Value**	**Function**
MODE	0	initialize PASS
	1	obtain next data record; it will not be modified
	2	obtain next data record; it may be modified
	3	terminate PASS (used only after MODE=1)

D.2.3 FINISH Record

One way to communicate the number of data records to NONMEM is to include this number in field 3 of the DATA control record. Another way is to simply place a blank or zero in that field and place a FINISH record after the last FORTRAN record in the input data file. The FINISH record is regarded as part of the input data file per se, rather than as another control record. Its format is quite simple, though quite different from that of the control records, and is given by the following two rules.

1. The FINISH record contains blank characters in positions 1-76 and 80 and the characters F, I and N in positions 77, 78 and 79, respectively.
2. If m is the number of FORTRAN records spanned per data record, there must be m-1 blank FORTRAN records inserted before the FINISH record.

In addition, for one to be able to use the FINISH record the following three rules must be followed.

1. All FORTRAN records of the input data file must be 80 characters long.
2. All FORTRAN records of the input data file except the FINISH record must have blank characters in positions 77-80.
3. The total number of A, B and F codes on the FORMAT record, including their multiplicities, must equal, i.e. not exceed, the number of data items specified on the DATA record.

The first of these three rules is always satisfied when the input data file is embedded in the control stream.

Use of the FINISH record is illustrated in Figure 10. The summary of the problem specification for this example is shown in Figure 13. There it may be seen that NONMEM has counted the number of data records and recorded this number on the summary.

The necessary change in the DATA record that accompanies the use of the FINISH record is reflected in the format for this record given in Table D.2.3.i.

Table D.2.3.i DATA record format ***(See Introduction to NONMEM VI March 2008 for current format.)***

Field no.	Value	Function
1	0 or blank between 20 & 99	data set embedded in the control stream FORTRAN unit number for data file
2	0 or blank 1	FORTRAN unit not to be rewound FORTRAN unit to be rewound
3	0 or blank positive integer	FINISH record used number of data records
4	between 1 & 20	number of data items per data record

D.2.4 Label Option

From Figure 13 it may be noted that the label of the prediction data item is PRD1, rather than the standard label, PRED. Nonstandard labeling for any one of the three data items: prediction, residual and weighted residual data items, is in general accomplished by including user-chosen labels for all three data items on the LABEL record. These three labels follow the labels that generally occur on the LABEL record. In addition, a one is placed in field 6 of the ITEM record. See Figure 10 for illustration. There the user-chosen labels for the residual and weighted residual data items coincide with the standard labels. (Nonstandard labeling, as described here, is not available in NONMEM version I.) The format for the ITEM record, showing the six fields so far defined, is given in Table D. 2.4.i.

Table D.2.4.i ITEM record format ***(See Introduction to NONMEM VI March 2008 for complete format.)***

Field no.	Value	Function
1	between 0 & 20 20	index of ID data item
2	between 1 & 20 20	index of DV data item
3	between 0 & 20	index of MDV data item
4	between 0 & 20	number of data item indices in INDXS
5	0 or blank 1	no user data item labels supplied user data item labels supplied
6	0 or blank 1	standard labels PRED, RES, WRES used nonstandard labels used

D.2.5 COVARIANCE Record

There are two matrices that are computed in the Covariance Step as intermediate steps toward computing the covariance matrix. The first matrix, the R matrix, is the Hessian matrix (or second derivative matrix) of the objective function, evaluated at the final estimate of the model parameter (θ,

Ω, Σ). The second matrix, the S matrix, is the sum of matrices, S_j, one matrix for each individual. Each matrix, S_j, is $\nabla_j \nabla'_j$, where ∇_j is the gradient (column) vector of the contribution to the objective function from the jth individual, evaluated at the final estimate of the model parameter Under the assumption that all random effects are normally distributed, the R and S matrices, each divided by the number, N, of individuals in the sample, tend to the same matrix as N increases. In this case the inverse of either matrix serves as an estimate of the true covariance matrix. When the normality assumption is not made, the matrix $R^{-1}SR^{-1}$ estimates the true covariance matrix. This is the default estimate in NONMEM.

The user may specify that the covariance matrix be given by R^{-1} or S^{-1}. (This feature is not available in NONMEM Version I, but in Version I the R and/or S matrices may be printed; see below.) In the first case, the standard errors are based on R^{-1}, the inverse covariance matrix is given by R, the correlation matrix is that of R^{-1}, and the eigenvalues (see section D.2.6) are those of R^{-1}. Similarly in the second case. To specify that the covariance matrix be given by R^{-1}, a one is placed in field 2 of the COVARIANCE record. To specify that the covariance matrix be given by S^{-1}, a two is placed in this field. To use the default covariance matrix, a blank or zero is placed in this field. (In NONMEM Version *I*, field 2 is used for a different purpose; see below.)

Suppose the default covariance matrix is used. Then either the R or S matrix, or both matrices, may be printed at the user's option. To specify that the R matrix, but not the S matrix, be printed, a one is placed in field 3 of the COVARIANCE record. To specify that the S matrix, but not the R matrix be printed, a two is placed in this field. To specify that both R and S be printed, a 3 is placed in this field. The problem specification in Figure 10 specifies that both matrices are to be printed. The desired printout is given in Figures 14 and 15.

If a blank or zero is placed in field 3, neither the R nor S matrix is printed (as long as there is also a blank or zero in field 2). Users of NONMEM Version I should use field 2 for the purpose of specifying printing of the R and/or S matrix, rather than field 3.

For a summary of the COVARIANCE record format see the next section.

D.2.6 Eigenvalues

Another statistic may be output at the user's option. This is the vector of eigenvalues of the correlation matrix. The individual eigenvalues are ordered from the least in value to the largest in value. To obtain the eigenvalues, a one is placed in field 4 of the COVARIANCE record. See Figure 10 for illustration. The eigenvalues for the example are given in Figure 16. The complete format of the COVARIANCE record is given in Table D.2.6.i.

Table D.2.6.i COVARIANCE record format ***(See Introduction to NONMEM VI March 2008 for current format.)***

Field no.	Value	Function
1	0 or blank 1 2	Covariance Step conditionally implemented Covariance Step unconditionally implemented Covariance Step omitted
If the value is 2, the subsequent fields may be ignored.		
2	0 or blank 1 2	covariance matrix set equal to $R^{-1}SR^{-1}$ covariance matrix set equal to R^{-1} covariance matrix set equal to S^{-1}
3	0 or blank 1 2 3	neither R nor S printed R matrix printed S matrix printed both R and S printed
4	0 or blank 1	eigenvalues not printed eigenvalues printed

Users of NONMEM Version I should use field 3 for the purpose of specifying printing of eigenvalues, rather than field 4.

D.3 Example Involving Complete Multivariability

D.3.1 Introduction

In the old example clearance and rate constant of elimination resulting from the same dose were assumed to be statistically independent conditional on the individual to whom the dose was given, i.e. conditional on the values of the first type random effects for the individual. It could not have been otherwise, since it is a property of the general model for one-level nested random effects as described in NONMEM Users Guide, Part I, that the joint levels of the second type random effects are taken to vary between observations and be independent. As a consequence, in the example, sigma was assumed to be diagonal. Since only one of ε_{1ij} and ε_{2ij} actually affect the observation, y_{ij}, these two random variables cannot be taken to be correlated so long as the above mentioned property of the general model must be satisfied. However, this property actually need not be satisfied, and this is the main idea to be discussed here and in subsequent sections. Let us first extend the model in the old example so that clearance and rate constant of elimination are not taken to be conditionally independent, and sigma is not taken to be diagonal.

Write y_{1ij} for the CL of the ith pair in the jth individual, and y_{2ij} for the KE of the ith pair in the jth individual. Also, let x_{11ij} and x_{12ij} both be the weight of the jth individual (for all i), and let x_{21ij} and x_{22ij} be the numbers 0 and 1, respectively for all i and j. Then let the model for y_{rij} be given by

$$y_{rij} = f(\theta_1, \theta_2, \theta_3, x_{1rij}, x_{2rij})$$

$$+g_1(x_{2rij})\eta_{1j} + g_2(x_{2rij})\eta_{2j}$$
$$+h_1(x_{2rij})\varepsilon_{1ij} + h_2(x_{2rij})\varepsilon_{2ij}$$

Where

$$f(\theta_1,\theta_2,\theta_3,x_1,x_2) = \{\theta_1,x_1+\theta_2 \text{ if } x_2=0$$
$$\{\theta_3 \text{ if } x_2=1$$

$$g_1(x_2)\begin{cases}1 & if\ x_2=0\\ 0 & if\ x_2=1\end{cases}$$

$$g_2(x_2)\begin{cases}1 & if\ x_2=0\\ 0 & if\ x_2=1\end{cases}$$

$$h_k(x_2) = g_k(x_2) \quad k=1,2$$
$$E(\eta_{1j},\eta_{2j})=0 \quad \mathrm{Cov}(\eta_{1j},\eta_{2j})=\Omega \quad for\ all\ j$$
$$E(\varepsilon_{1ij},\varepsilon_{2ij})=0 \quad \mathrm{Cov}(\varepsilon_{1ij},\varepsilon_{2ij})=\Sigma \quad for\ all\ i,j\ .$$

There are several differences between this model formulation and the old one. First, and least important, the argument lists for the g and h functions have been shortened since these functions never really depended on those arguments that have been deleted. Second, the second type random effects do not assume different values with different observations; they may assume different values with different pairs of observations. This is why the observations have been triply subscripted, whereas before they were doubly subscripted. However, it is still assumed that the different joint levels of the second type random effects, i.e. the vectors ($\varepsilon_{1ij}, \varepsilon_{2ij}$) for all i and j, are statistically independent. As a result of this difference in the model formulation, y_{1ij} and y_{2ij} are certainly not conditionally independent. Third, sigma is not constrained to be diagonal. It need not be since, although only ε_{1ij} affects y_{1ij} and only ε_{2ij} affects y_{2ij}, as just observed y_{1ij} and y_{2ij} are not conditionally independent.

This model is but an example of a more general one that may indeed be implemented with NONMEM, and this general model is described in the next section, D.3.2. To use a triply subscripted model, the so-called level two data item must be used, and this is discussed and illustrated for the example in section D.3.3.

D.3.2 The General Model with One-Level Vested Random Effects

The complete form of the general model with one-level nested effects that is implemented in NONMEM is given in this section. The special case, given in section F.5 of the NONMEM Users Guide, Part I, is still quite general and should suffice for most purposes.

The observations are triply subscripted: y_{rij}, r=1,m$_{ij}$, i=1,n$_j$, j=1,J where for some j, h$_j$ >1 and for some i, j, m$_{ij}$ >1. The model for y_{rij} is given by

$$y_{\mathrm{rij}} = f_{rij}(\,\theta\,) + \sum_{k=1}^{K} g_{krij}(\theta)\,\eta_{kj} + \sum_{l=1}^{L} h_{lrij}(\,\theta\,)\,\varepsilon_{lij}$$

E$(\eta_{1j}, \eta_{2j}, ..., \eta_{Kj}) = 0$

for all j

Cov $(\eta_{1j}, \eta_{2j}, ..., \eta_{Kj}) = \Omega$

E$(\varepsilon_{1ij}, \varepsilon_{2ij} ... \varepsilon_{Lij}) = 0$

for all i, j

Cov $((\varepsilon_{1ij}, \varepsilon_{2ij} ... \varepsilon_{Lij}) = \Sigma$

where the f_{rij}, g_{krij}, and h_{lrij} are functions of the vector-valued parameter, θ, the vectors $\eta_j = (\eta_{1j}, \eta_{2j}, ..., \eta_{Kj})$ are independent random vectors, the vectors $\varepsilon_{ij} = \varepsilon_{1ij}, \varepsilon_{2ij} ... \varepsilon_{Lij})$ are independent random vectors, and for each i,j, the covariance of η_j and ε_{ij} is zero. The triple subscripting on the function f, and on the functions g_k and h_l, serves as a shorthand alternative to listing the values of the x's as was done in the example. This model looks very similar to the one in NONMEM Users Guide, Part I. Formally, the only difference appears to be the use of triple, rather than double, subscripting. But this difference is nonetheless the important distinction to be made here and is related to the full multivariate character of observations from a given individual, conditional on the individual.

For θ, and r, i, j, the values of $g_{1rij}(\theta)$, $g_{2rij}(\theta), ... g_{Krij}(\theta)$ are returned in PRED in G(1), G(2),...,G(K), respectively, and the values of $h_{1rij}(\theta)$, $h_{2rij}(\theta), ... h_{Lrij}(\theta)$ are returned by PRED in H(1), H(2),…, H(L) respectively. Both K and L can be at most 5.

The (default) objective function for this model is

$$O(\theta, \Omega, \Sigma) = \sum_{j=1}^{J} [\log \det C_j(\theta, \Omega, \Sigma) + R_j(\theta, \Omega, \Sigma) R'_j(\theta, \Omega, \Sigma)]$$

where

$$y_j = (y_{11j}, y_{21j}, ..., y_{m_{1j}1j}, ..., Y_{ym_{n_j j} n_j j})$$

$C_j(\theta, \Omega, \Sigma) =$ covariance matrix of y_j under the model

$$f_j(\theta) = (f_{11j}(\theta), f_{21j}(\theta), ..., f_{m_{1j}1j}(\theta), ..., f_{m_{n_j j} n_j j}(\theta))$$

$$R_j(\theta, \Omega, \Sigma) = (Y_j - f_j(\theta)) C_j(\theta, \Omega, \Sigma)^{-1/2}.$$

Now let

$$g_{kj}(\theta) = (g_{k11j}(\theta), g_{k21j}(\theta), ..., g_{km_{1j}1j}(\theta), ... g_{km_{n_j j} n_j j}(\theta))$$

and

$$h_{lj}(\theta) = (h_{l11j}(\theta), h_{l21j}(\theta), ..., h_{lm_{1j}1j}(\theta), ..., h_{lm_{n_j j} n_j j}(\theta)$$

Let $G_j(\theta)$ be the *K* by Q_j matrix whose *k*th row is $g_{kj}(\theta)$, and let $H_j(\theta)$ be the *L* by Q_j matrix whose *l*th row is $h_{lj}(\theta)$. Here

$Q_j = m_{1j} + m_{2j} + ... + m_{n_j j}$.

The matrix C_j can be expressed in terms of the matrices G_j and H_j. To do this some further notation is needed.

Let Q be a positive integer, and let there be a vector of positive integers, $q = (q_1, q_2, ...q_n)$, whose sum is Q. Let $A = (a_{ij})$ be a Q-dimensional matrix. Define diag$_q A$ to be the block diagonal matrix

$$\begin{pmatrix} A_1 & & & 0 \\ & \cdot & & \\ & & \cdot & \\ & & & \cdot \\ 0 & & & A_n \end{pmatrix}$$

Where A_k is the submatrix of A consisting of those elements a_{ij} such that

$$\sum_{l=1}^{k-1} q_l < i, j < \sum_{l=1}^{k} q_l \ .$$

Using this notation, we have

$$C_j(\theta, \Omega, \Sigma) = G_j'(\theta)\Omega G_j(\theta) + diag_{q_j}(H_j'(\theta)\sum H_j(\theta))$$

where $q_j = (m_{1j}, m_{2j}, ..., m_{n_j j})$.

D.3.3 The Level Two Data Item

To use the triply subscripted model the groups of observations in which the second type random effects do not vary must be delineated for NONMEM. This is done analogously to the way the groups of observations in which the first type random effects do not vary are delineated. That is, a data item like the ID data item is used. One might also call the first type of random effects, level one random effects and the second type of random effects, level two random effects. (There are two levels of random effects, but there is only one level of nesting.) Then the ID data item might be also called the level one (L1) data item, since it is used to group observations according to level one random effects. The data item to be described in this section is called the level two (L2) data item since it is used to group observations according to level two random effects.

In the general model the dependent observations are partitioned into groups in which the level two effects do not vary. These groups are called level two units. In NONMEM the data records whose DV data items are those of a given level two unit may be grouped together, and this is accomplished in part by including in each data record an L2 data item identifying the level two unit containing the DV data item in the record. The L2 data items need not be integer valued (in contrast to the ID data items). A problem specification for the example is given in Figure 17. There the fifth data item of each data record is a L2 data item. Grouping is effected by including the L2 data items in the data records and by arranging that all data records with the same L2 data item be contiguous. The data records with the same L2 data item are collectively called a level two unit record. As with L1 data items, the same L2 data item may actually be used to identify different level two unit records, providing these records are not contiguous in the data set. Level two unit records may consist of one or more case records (see section B.1, NONMEM Users Guide, Part I). Only the last data record of a case record need contain an L2 data item. Obviously, a level two unit record should be contained in a single individual record.

Lastly, the index of the L2 data items should be placed in field 7 of the ITEM record. See Figure 17 for illustration. The complete format of the ITEM record is given in Table D.3.3.i.

Table D.3.3.i ITEM record format

Field no.	Value	Function
1	between 0 & 20	index of ID data item
2	between 1 & 20	index of DV data item
3	between 0 & 20	index of MDV data item
4	between 0 & 20	number of data item indices in INDXS
5	0 or blank 1	no. user data item labels supplied User data item labels supplied
6	0 or blank 1	standard labels PRED, RES, WRES used nonstandard labels used
7	between 0 & 20	index of L2 data item

Users of NONMEM Version I should use field 6 of the ITEM record for the purpose of specifying the presence of L2 data items. (This feature is not available in NONMEM Version I Level 1.)

The code for PRED for the example remains unchanged. In addition to inclusion of L2 data items, the other important difference in the problem specification from that of the old example is the inclusion of an initial estimate for the off-diagonal element of sigma. The initial estimate of sigma (as a matrix) is obtained by first obtaining from each individual an estimate of the covariance matrix of his CL and KE in the usual way. Then these individual estimates are averaged to obtain an estimate, which after being rounded to the first significant figure, serves as the initial estimate of sigma.

There is little difference between the final estimate, as given in Figure 18, and that of the old example. The estimated intraindividual correlation between CL and KE, though, is .54. The standard error of the final estimate is given in Figure 19, and the standard error of the estimate of the intraindividual covariance between CL and KE, in particular, supports the significance of the correlation.

D.4 Example Involving Utilities CHOL and MULT

D.4.1 introduction

In section D.2, a NONMEM utility subroutine, PASS, is described. This subroutine can be called by PRED. In this section two other NONMEM utility subroutines that can be called by PRED are described.

The first utility, CHOL, allows the Cholesky square root of a positive definite matrix to be obtained in a convenient way. (CHOL is not available with NONMEM Version I Level 1.) The second utility, MULT, allows the two terms

$$\log\det C_j(\theta,\Omega,\Sigma) \text{ and } R_j(\theta,\Omega,\Sigma)R_j'(\theta,\Omega,\Sigma)$$

occurring in the (default) objective function to be weighted unequally, where the weights are computed in PRED and may vary with j. MULT can also be used with CRIT (see section C.6). (MULT is not available with NONMEM Version I.) These utilities may be used for a number of different reasons. However, with this example we illustrate only one way in which they may be used. The theoretical aspects of the example are given below. CHOL is described in section D.4.2, and MULT is described in section D.4.3. The user may wish to go directly to these sections to obtain these descriptions and skip the example.

Consider again the data used in the example of section D.3 and the model for this data developed there. For each j, estimates of the three quantities

$$CL_j = \theta_1 w_j + \eta_{1j}$$

$$KE_j = \theta_3 + \eta_{2j}$$

and Σ, where w_j is the weight of the jth subject, may be obtained from just the 12 observations in the *j*th subject. $\hat{C}L_j$ is given by the sample mean of the 6 clearances: $\hat{K}E_j$ is given by the sample mean of the 6 rate constants: $\hat{\Sigma}$ is given by the sample covariance matrix of the 6 pairs of clearance and rate constant. Write $\hat{\Sigma}_j$, instead of $\hat{\Sigma}$, to indicate that the estimate is based only on the observations from the jth subject. Now assume that the second type random effects are normally distributed. Let *m*=6. Then conditional on η_{1j} and η_{2j}, $(\hat{C}L_j,\hat{K}E_j)$ is multivariate normally distributed with mean (CL_j,KE_j) and covariance Σ/m, while $(m-1)\hat{\Sigma}_j$ is independently Wishart distributed with parameters Σ and m-1 (2). So unconditionally, $(\hat{C}L_j,\hat{K}E_j)$ is multivariate normally distributed with mean $(\theta_1 w_j,\theta_3)$ and covariance $\Omega+\Sigma/m$, while $(m-1)\hat{\Sigma}_j$ is independently Wishart distributed with parameters Σ and m-1. Therefore, the negative log-likelihood of (θ,Ω,Σ) based on the $\hat{C}L_j,\hat{K}E_j$, and $\hat{\Sigma}_j$ may be written in a straight forward manner and is the sum of two components, one component, S_{1j}, being the contribution from $(\hat{C}L_j,\hat{K}E_j)$ and the other component, S_{2j}, being the contribution from $\hat{\Sigma}_j$. The negative log likelihood of (θ,Ω,Σ) based on $\hat{C}L_j,\hat{K}E_j$ and $\hat{\Sigma}_j$ for all j, is of course the sum of $S_{1j}+S_{2j}$ over all j. We shall indicate how NONMEM may be used to obtain the maximum likelihood estimate of (θ,Ω,Σ) from the $\hat{C}L_j, \hat{K}E_j$. and $\hat{\Sigma}_j$. It may be shown that theoretically this estimate coincides with the estimate obtained in section D.3, and that estimate is consistent even though it is developed without the above normality assumption on the second type random effects.

The first component, S_{1j}, is

$$\log\det C_j + ((y_j - f_j)C_j^{-1}(y_j - f_j)')$$

Up to an additive constant, where

$$C_j = \Omega + \Sigma / m$$

$$y_j = (y_{1j}, y_{2j}) \qquad y_{1j} = \hat{C}L_j \qquad y_{2j} = \hat{K}E_j$$

$$f_j = (f_{1j}, f_{2j}) \qquad f_{1j} = \theta_1 w_j \qquad f_{2j} = \theta_3$$

This has the form of a term of the NONMEM (default) objective function for a model with equivalently nested random effects (see section D.6 of NONMEM users Guide, Part I). In fact, with appropriate definitions for the functions g_{kij} and the omega matrix, in NONMEM an individual can be specified whose observations and predictions are y_j and f_j and whose C_j is $\Omega + \Sigma / m$. Define the omega matrix to be the block diagonal matrix

$$\begin{pmatrix} \Omega & 0 \\ 0 & \Sigma \end{pmatrix}$$

(use the constraint on omega described in section B). Define

$g_{1ij} = 1$ if i=1
$\quad 0$ otherwise
$g_{2ij} = 1$ if i=2
$\quad 0$ otherwise
$g_{3ij} = C$ if i=1
$\quad 0$ otherwise
$g_{4ij} = C$ if i=2
$\quad 0$ otherwise

where $c = 1/\sqrt{6}$. That is, G_j' is

$$\begin{pmatrix} 1 & 0 & C & 0 \\ 0 & 1 & 0 & C \end{pmatrix}$$

The second component, S_{2j}, is

$$(m-1)\log\det\Sigma + (m-1)tr(\hat{\Sigma}_j\Sigma^{-1})$$

up to an additive constant. Since $\hat{\Sigma}_j$ is positive definite, there exists a 2-dimensional lower triangular matrix T_j such that $T_jT_j^{\prime} = \hat{\Sigma}_j$. This matrix is called the Cholesky square root of $\hat{\Sigma}_j$. Since

$$tr(\hat{\Sigma}_j\Sigma^{-1}) = tr(T_j\Sigma^{-1}T_j^{\prime}),$$

the second component may be written as the sum of two terms:

$$((m-1)/2)\log\det\Sigma + (m-1)T_{jr}\Sigma^{-1}T_{jr}^{\prime} \qquad r = 1,2$$

where T_{jr} is the rth row of T_j. The rth term may in turn be written as

$$((m-1)/2)\log\det C_j + (m-1)(y_j - f_j)C_j^{-1}(y_j - f_j)'$$

where now

$$C_j = \Sigma \qquad y_j = T_{jr} \qquad f_j = 0.$$

Momentarily ignoring the two coefficients $(m-1)/2$ and $m-1$, this again has the form of a ten of the NONMEM (default) objective function for a model with equivalently nested random effects. With appropriate definitions for the functions g_{kij} and with the <u>same</u> block diagonal omega matrix as given above, in NONMEM an individual can be specified whose observations and predictions are y_j and f_j and whose C_j is Σ. Define

$g_{1ij} = 0$ for i=1,2

$g_{2ij} = 0$ for i=1,2

$g_{3ij} = 1$ If i=1
0 if i=2

$g_{4ij} = 0$ if i=1
1 if i=2

That is, $G_j^{\prime}$ is

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Hence, three individuals may be specified, each of which contributes a term to the NONMEM objective function, and the sum of these three terms is $S_{1j} + S_{2j}$. Thus NONMEM may be used to obtain the desired maximum likelihood estimate. There remain, though, two problems. The first is that of obtaining the T_{jr}, and the second is that of introducing the coefficients, $(m-1)/2$

and $m-1$, into the objective function where this is necessary. Solutions to these problems are presented in the next two sections.

D.4.2 Cholesky Square Root (CHOL)

CHOL is a NONMEM utility subroutine that may be called by PRED. It computes the Cholesky square root of a given positive definite matrix A, i.e. the lower triangular matrix B satisfying BB'=A. (It is not available with NONMEM Version I Level 1.) It is used in a fashion that is particularly convenient for PRED. If the dimension of A is n, then CHOL is called n+1 times in succession. The first call initializes the routine. On the (i+1)st call, the ith row of A is passed to CHOL, and CHOL returns the ith row of B (i=1,...,n). Now suppose A is stored in the data set with the elements of each row of A stored as data items of some data record, a different data record for each row. Until each of the involved data records is passed to PRED, the entire matrix A cannot be made available to PRED. Nonetheless, due to the way in which CHOL is organized (as just described), as each data record is passed to PRED a different row of B can be made available to PRED.

There are four arguments in the subroutine's argument list. The first argument should be set by PRED to the integer 0 or 1 according to whether the call to CHOL is the initialization call or not. The second argument should be set to the (integer) dimension of A. The third argument should be set to the ith row vector of A on the (i+1)st call to CHOL. (The elements a_{ij}, j>i, are ignored.) Upon return to PRED, the ith row vector of B is found in the third argument. (The elements b_{ij}, j>i, are all zero.) The fourth argument contains an integer return code from CHOL each time a return occurs. Upon return from the initialization call, this code is always 0. Upon return from the (i+1)st call, this code is 0 unless CHOL has determined that based on the first i rows of A, A is algorithmically not positive definite, in which case the return code is 1. At the initialization call, the fourth argument should be set by PRED to either 0 or 1. If it is set to 0 at subroutine initialization, then whenever CHOL is about to return with a return code of 1, CHOL actually terminates NONMEM execution and prints an appropriate message (so the return never really occurs). If the fourth argument is set to 1 at subroutine initialization, then CHOL always returns control to PRED along with the appropriate return code. These arguments are summarized in Table D.4.2.i.

Table D.4.2.i Arguments of CHOL

Argument	Value	Function
MODE	0 or blank 1	initialize CHOL a matrix row is being passed
N	between 1 & 5	dimension of matrix
R		row of matrix
IER	upon input when MODE is 0:	
	0 1	CHOL terminates execution if matrix is algorithmically not positive definite CHOL will always return control to PRED
	upon output:	
	0 1	normal return matrix is algorithmicallynot positive definite

For the example, the code for PRED is shown in Figure 20, and the problem specification is shown in Figure 21. Look first at the input data file. The data records are arranged in 12 groups of six data records each. (Each data record of a particular group has the same first digit in its ID data item.) The DV data item of the first data record of the jth group is $\hat{C}L_j$, and the DV data item of the second data record of the jth group is $\hat{K}E_j$. The DV data items of the remaining 4 data records of the jth group are all zero. However, the fifth data item of the third data record of the jth group is the first element of the first row of $\hat{\Sigma}_j$, and the fifth and sixth data items of the fifth data record of the jth group are the first and second elements of the second row of $\hat{\Sigma}_j$. Therefore, the rows of T_j may be obtained. This is accomplished by using CHOL during the problem initialization phase of PRED, so that these rows may be stored as DV data items to be used by NONMEM during problem execution. (See the discussion below about the PRED code.) The first and second elements of T_{j1} may be obtained from the third data record of the jth group, and these elements are to be stored as the DV data items of the third and fourth data records, respectively. The first and second elements of T_{j2} may be obtained from the fifth data record of the jth group, and these elements are to be stored as the DV data items of the fifth and sixth data records, respectively.

Each of the groups of six data records just described is comprised of three individual records. Consider the jth group. The DV data items of the first individual record are $\hat{C}L_j$ and $\hat{K}E_j$. The DV data items of the second individual record are the elements of T_{j1}, and the DV data items of the third individual record are the elements of T_{j2} (after PRED initialization). These three individual records have the three ID data items jl, j2 and j3, respectively. A type data item also appears in each data record, and this data item takes a unique value for each data record of an individual record so that these different data records may be clearly identified. This is the fourth data item in the data record. The weight data item appears as the second data item in both data records of the first individual record of the jth group.

Notice too that, whereas in the previous examples where clearance has been modeled as a linear function of weight, whose intercept has been constrained to be zero, in this example the intercept has simply been eliminated so that there are only two elements of theta. These are the slope parameter of the linear function and the population mean elimination rate constant. Omega is constrained to be block diagonal with two blocks, each of dimension 2. Notice the use of the structure and BLOCK SET records to accomplish this (see section B).

Regarding next the code for PRED, it may be seen that there are three different pieces of code, starting at statements 5, 10 and 25, only one of which is executed during any particular call to PRED, depending on the value of ICALL. During problem initialization, i.e. when ICALL is 1, CHOL is used to obtain the T_j, as mentioned above. This task is embedded in a pass through the data set using the NONMEM utility PASS (see section D.2.2). For each j, the 6 data records of the jth group are successively passed to PRED. The first two data records are simply ignored.

Then CHOL is initialized. Then during the first of the two executions of the DO loop, CHOL is used to obtain the elements of T_{j1} from the information on the third data record, and these elements are stored as the DV data items of the third and fourth data records. During the second execution of the DO loop, CHOL is used to obtain the elements of T_{j2} from the information on the fifth data record, and these elements are stored as the DV data items of the fifth and sixth data records. When ICALL is 2, F and G are computed according to the discussion in section D.4.1. See the next section for the explanation of the call to MULT.

D.4.3 Reweighting Objective Function (MULT)

The (default) NONMEM objective function is the sum of terms

$$A_j \log\det C_j(\theta,\Omega,\Sigma) + B_j R_j(\theta,\Omega,\Sigma) R_j^{\prime}(\theta,\Omega,\Sigma)$$

over all j, where all A_j and B_j are equal to 1. Other values of A_j and B_j may be used. The NONMEM utility subroutine MULT allows these quantities to be set by PRED to values other than 1. (This feature is not available in NONMEM Version I.) To set A_j and B_j to a and b, say, PRED issues a single call to MULT during any one or more calls to it from NONMEM where i) ICALL is 2 and ii) some data record from the jth individual record is in DATREC. (At least one such call to MULT must be issued while ICALL is 2, and data records from the jth individual record are being passed to PRED in DATREC.) MULT has only two arguments; the first is set to a and the second to b. (If during each of a number of calls to PRED while i and ii hold, MULT is called, then each time MULT is called it should be called with the same values for its two arguments.)

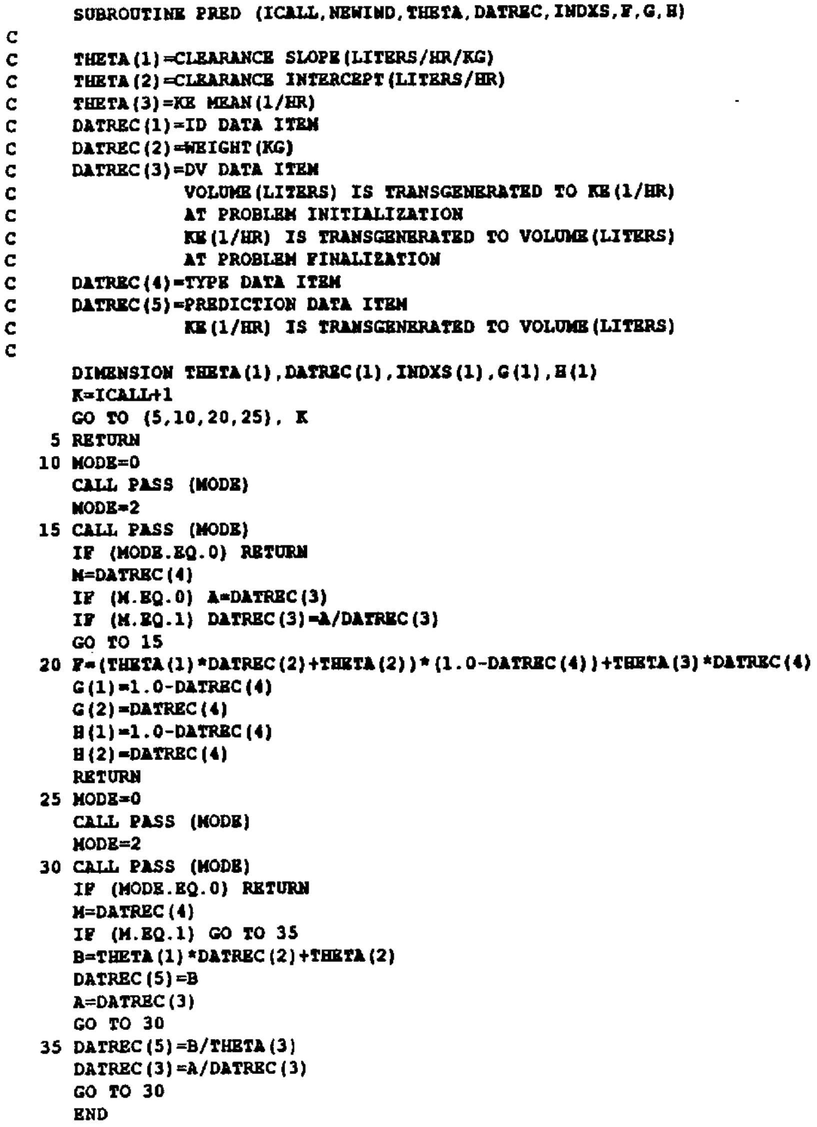
In section C.6 it is explained that the contribution to the objective function from the jth individual may be taken to be

$$A_j \log\det C_j(\theta,\Omega,\Sigma) + B_j \varphi_j(R_j(\theta,\Omega,\Sigma))$$

where A_j and B_j are equal to 1 and φ_j is some function of the vector of weighted residuals from the jth individual other than the inner product. MULT can be used, as described above, to allow PRED to set A_j and B_j to values other than 1.

In the code for PRED in the example, given in Figure 20, it may be seen that with every individual record with ID data item ending in 2 or 3, MULT is called with arguments equal to $(m-1)/2$ and $m-1$ whenever the last data record of the individual record is in DATREC.

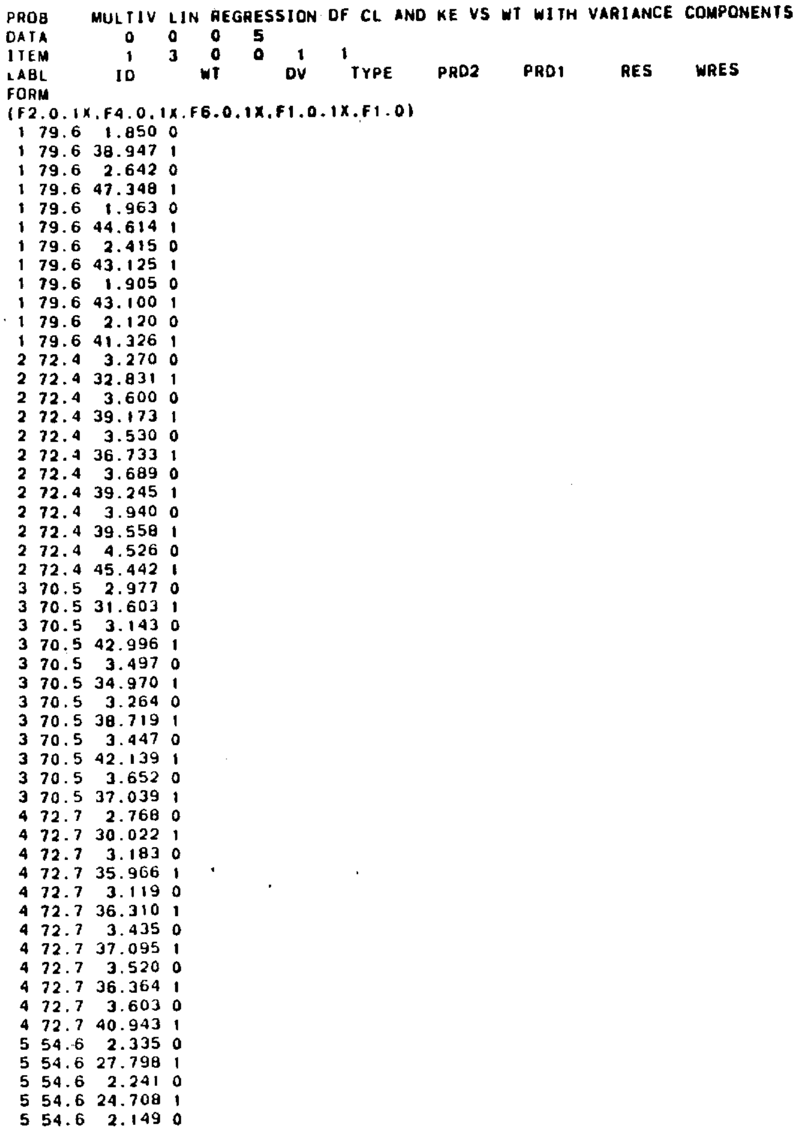
Figure 9



```
      SUBROUTINE PRED (ICALL,NEWIND,THETA,DATREC,INDXS,F,G,H)
C
C     THETA(1)=CLEARANCE SLOPE(LITERS/HR/KG)
C     THETA(2)=CLEARANCE INTERCEPT(LITERS/HR)
C     THETA(3)=KE MEAN(1/HR)
C     DATREC(1)=ID DATA ITEM
C     DATREC(2)=WEIGHT(KG)
C     DATREC(3)=DV DATA ITEM
C               VOLUME(LITERS) IS TRANSGENERATED TO KE(1/HR)
C               AT PROBLEM INITIALIZATION
C               KE(1/HR) IS TRANSGENERATED TO VOLUME(LITERS)
C               AT PROBLEM FINALIZATION
C     DATREC(4)=TYPE DATA ITEM
C     DATREC(5)=PREDICTION DATA ITEM
C               KE(1/HR) IS TRANSGENERATED TO VOLUME(LITERS)
C
      DIMENSION THETA(1),DATREC(1),INDXS(1),G(1),H(1)
      K=ICALL+1
      GO TO (5,10,20,25), K
    5 RETURN
   10 MODE=0
      CALL PASS (MODE)
      MODE=2
   15 CALL PASS (MODE)
      IF (MODE.EQ.0) RETURN
      M=DATREC(4)
      IF (M.EQ.0) A=DATREC(3)
      IF (M.EQ.1) DATREC(3)=A/DATREC(3)
      GO TO 15
   20 F=(THETA(1)*DATREC(2)+THETA(2))*(1.0-DATREC(4))+THETA(3)*DATREC(4)
      G(1)=1.0-DATREC(4)
      G(2)=DATREC(4)
      H(1)=1.0-DATREC(4)
      H(2)=DATREC(4)
      RETURN
   25 MODE=0
      CALL PASS (MODE)
      MODE=2
   30 CALL PASS (MODE)
      IF (MODE.EQ.0) RETURN
      M=DATREC(4)
      IF (M.EQ.1) GO TO 35
      B=THETA(1)*DATREC(2)+THETA(2)
      DATREC(5)=B
      A=DATREC(3)
      GO TO 30
   35 DATREC(5)=B/THETA(3)
      DATREC(3)=A/DATREC(3)
      GO TO 30
      END
```

Figure 9

Figure 10a



```
PROB    MULTIV LIN REGRESSION OF CL AND KE VS WT WITH VARIANCE COMPONENTS
DATA       0   0   0   5
ITEM       1   3   0   0   1   1
LABL      ID      WT      DV     TYPE     PRD2     PRD1      RES     WRES
FORM
(F2.0.1X.F4.0.1X.F6.0.1X.F1.0.1X.F1.0)
 1 79.6  1.850 0
 1 79.6 38.947 1
 1 79.6  2.642 0
 1 79.6 47.348 1
 1 79.6  1.963 0
 1 79.6 44.614 1
 1 79.6  2.415 0
 1 79.6 43.125 1
 1 79.6  1.905 0
 1 79.6 43.100 1
 1 79.6  2.120 0
 1 79.6 41.326 1
 2 72.4  3.270 0
 2 72.4 32.831 1
 2 72.4  3.600 0
 2 72.4 39.173 1
 2 72.4  3.530 0
 2 72.4 36.733 1
 2 72.4  3.689 0
 2 72.4 39.245 1
 2 72.4  3.940 0
 2 72.4 39.558 1
 2 72.4  4.526 0
 2 72.4 45.442 1
 3 70.5  2.977 0
 3 70.5 31.603 1
 3 70.5  3.143 0
 3 70.5 42.996 1
 3 70.5  3.497 0
 3 70.5 34.970 1
 3 70.5  3.264 0
 3 70.5 38.719 1
 3 70.5  3.447 0
 3 70.5 42.139 1
 3 70.5  3.652 0
 3 70.5 37.039 1
 4 72.7  2.768 0
 4 72.7 30.022 1
 4 72.7  3.183 0
 4 72.7 35.966 1
 4 72.7  3.119 0
 4 72.7 36.310 1
 4 72.7  3.435 0
 4 72.7 37.095 1
 4 72.7  3.520 0
 4 72.7 36.364 1
 4 72.7  3.603 0
 4 72.7 40.943 1
 5 54.6  2.335 0
 5 54.6 27.798 1
 5 54.6  2.241 0
 5 54.6 24.708 1
 5 54.6  2.149 0
```

Figure 10a

Figure 10b

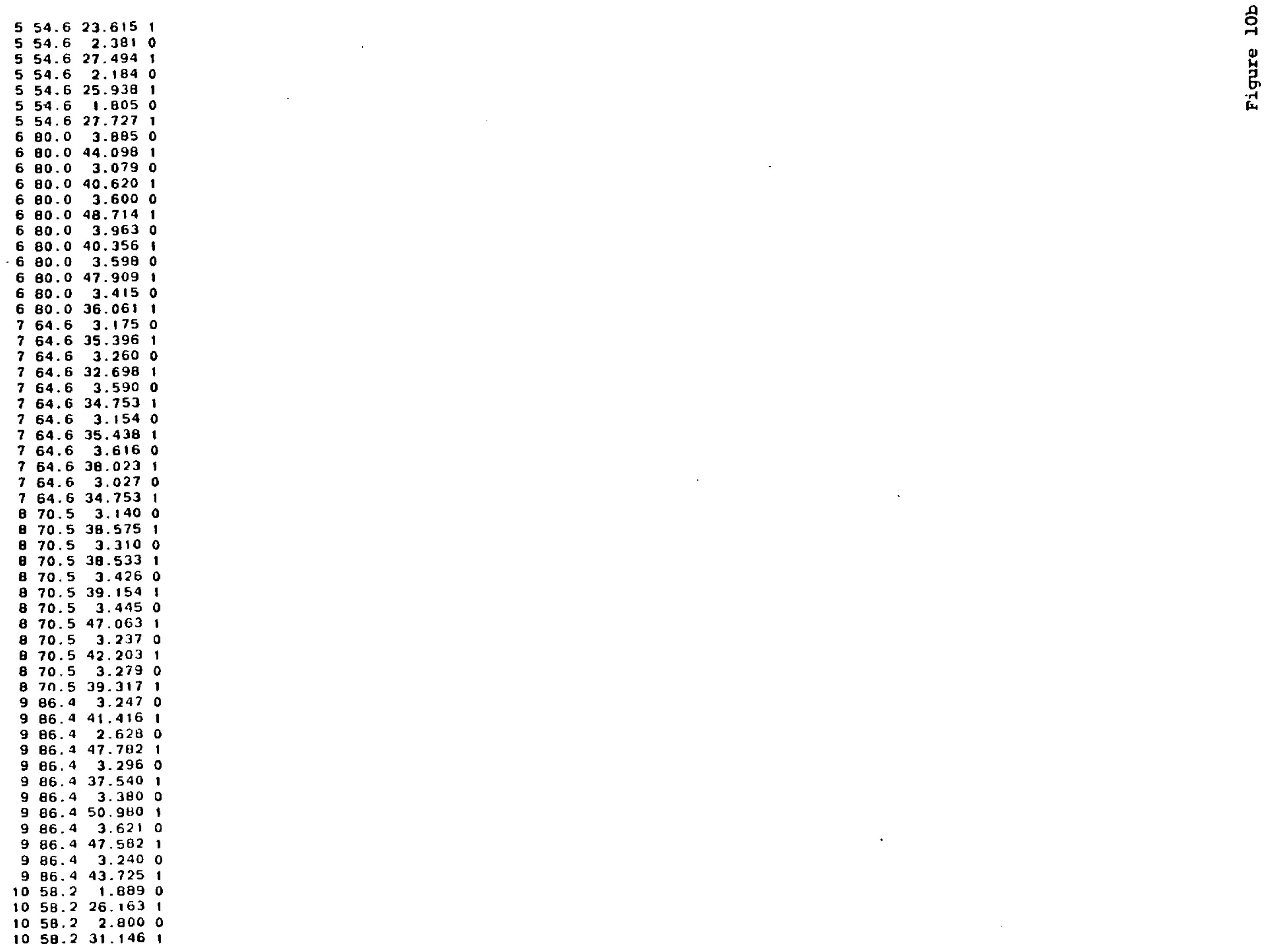
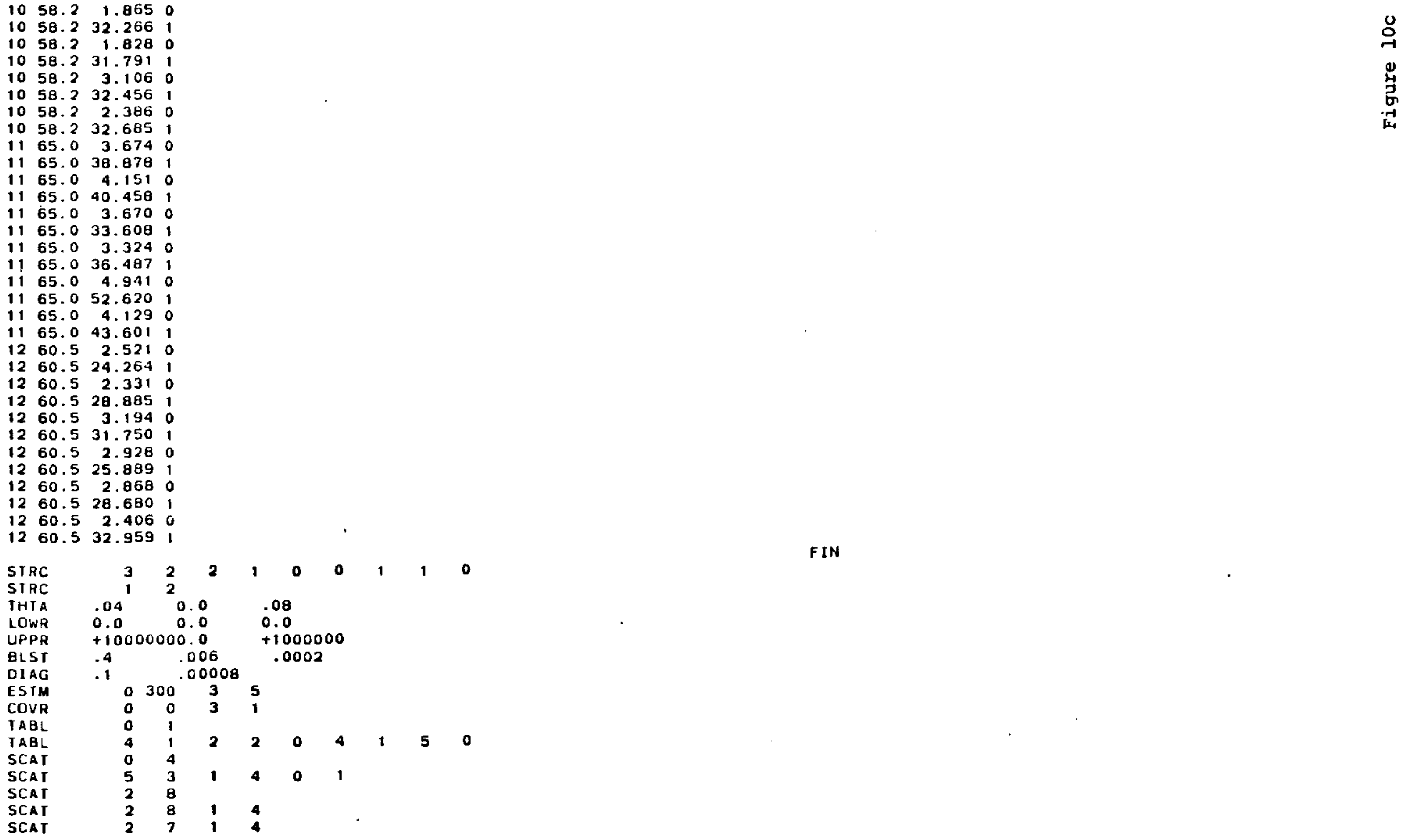


Figure 10b

```
 5 54.6 23.615 1
 5 54.6  2.381 0
 5 54.6 27.494 1
 5 54.6  2.184 0
 5 54.6 25.938 1
 5 54.6  1.805 0
 5 54.6 27.727 1
 6 80.0  3.885 0
 6 80.0 44.098 1
 6 80.0  3.079 0
 6 80.0 40.620 1
 6 80.0  3.600 0
 6 80.0 48.714 1
 6 80.0  3.963 0
 6 80.0 40.356 1
 6 80.0  3.598 0
 6 80.0 47.909 1
 6 80.0  3.415 0
 6 80.0 36.061 1
 7 64.6  3.175 0
 7 64.6 35.396 1
 7 64.6  3.260 0
 7 64.6 32.698 1
 7 64.6  3.590 0
 7 64.6 34.753 1
 7 64.6  3.154 0
 7 64.6 35.438 1
 7 64.6  3.616 0
 7 64.6 38.023 1
 7 64.6  3.027 0
 7 64.6 34.753 1
 8 70.5  3.140 0
 8 70.5 38.575 1
 8 70.5  3.310 0
 8 70.5 38.533 1
 8 70.5  3.426 0
 8 70.5 39.154 1
 8 70.5  3.445 0
 8 70.5 47.063 1
 8 70.5  3.237 0
 8 70.5 42.203 1
 8 70.5  3.279 0
 8 70.5 39.317 1
 9 86.4  3.247 0
 9 86.4 41.416 1
 9 86.4  2.628 0
 9 86.4 47.782 1
 9 86.4  3.296 0
 9 86.4 37.540 1
 9 86.4  3.380 0
 9 86.4 50.980 1
 9 86.4  3.621 0
 9 86.4 47.582 1
 9 86.4  3.240 0
 9 86.4 43.725 1
10 58.2  1.889 0
10 58.2 26.163 1
10 58.2  2.800 0
10 58.2 31.146 1
```

Figure 10c



```
10 58.2  1.865 0
10 58.2 32.266 1
10 58.2  1.828 0
10 58.2 31.791 1
10 58.2  3.106 0
10 58.2 32.456 1
10 58.2  2.386 0
10 58.2 32.685 1
11 65.0  3.674 0
11 65.0 38.878 1
11 65.0  4.151 0
11 65.0 40.458 1
11 65.0  3.670 0
11 65.0 33.608 1
11 65.0  3.324 0
11 65.0 36.487 1
11 65.0  4.941 0
11 65.0 52.620 1
11 65.0  4.129 0
11 65.0 43.601 1
12 60.5  2.521 0
12 60.5 24.264 1
12 60.5  2.331 0
12 60.5 28.885 1
12 60.5  3.194 0
12 60.5 31.750 1
12 60.5  2.928 0
12 60.5 25.889 1
12 60.5  2.868 0
12 60.5 28.680 1
12 60.5  2.406 0
12 60.5 32.959 1
                                                  FIN
STRC       3   2   2   1   0   0   1   1   0
STRC       1   2
THTA    .04       0.0       .08
LOWR    0.0       0.0       0.0
UPPR    +10000000.0       +1000000
BLST    .4         .006       .0002
DIAG    .1         .00008
ESTM       0 300   3   5
COVR       0   0   3   1
TABL       0   1
TABL       4   1   2   2   0   4   1   5   0
SCAT       0   4
SCAT       5   3   1   4   0   1
SCAT       2   8
SCAT       2   8   1   4
SCAT       2   7   1   4
```

Figure 10c

Figure 11

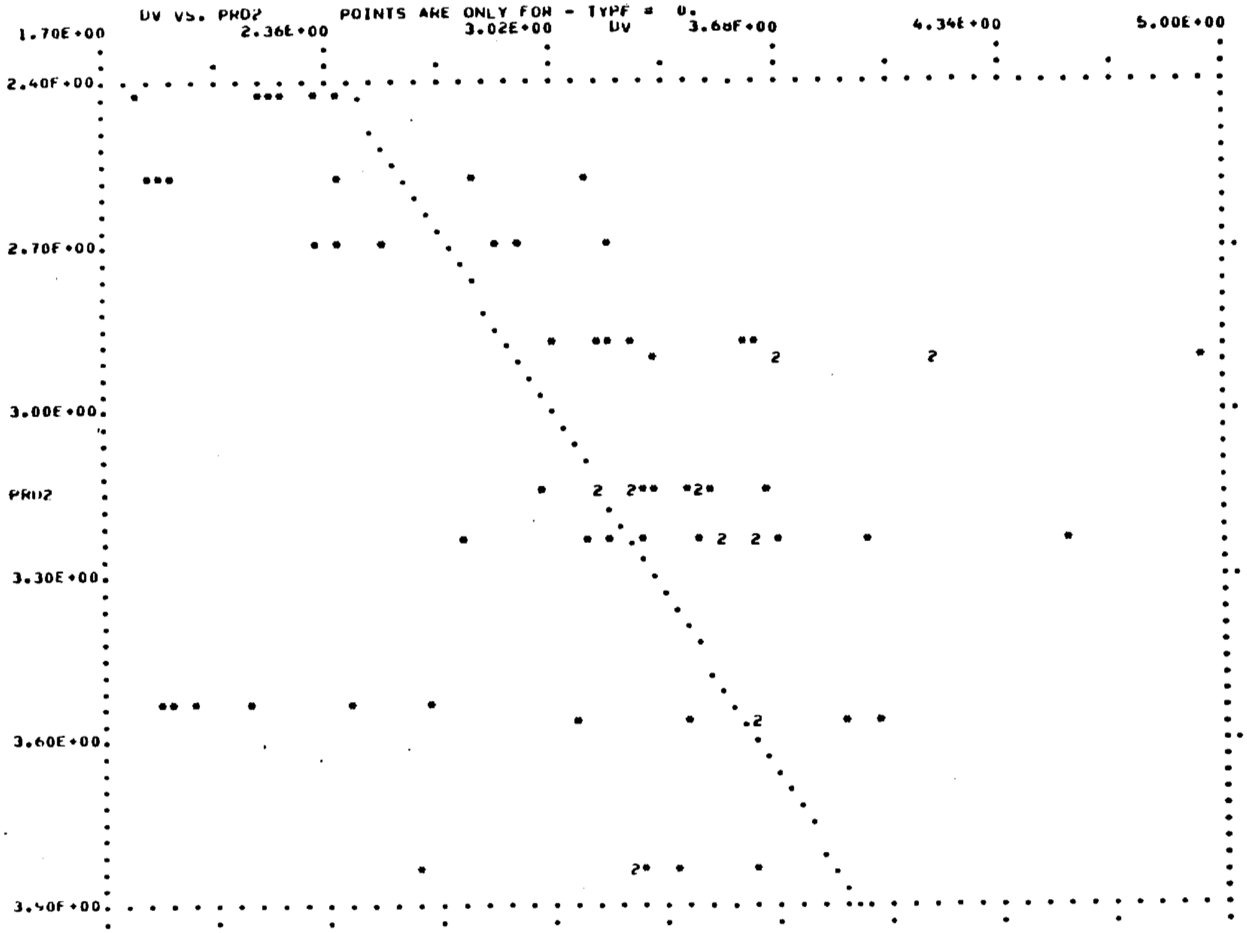


Figure 11

Figure 12

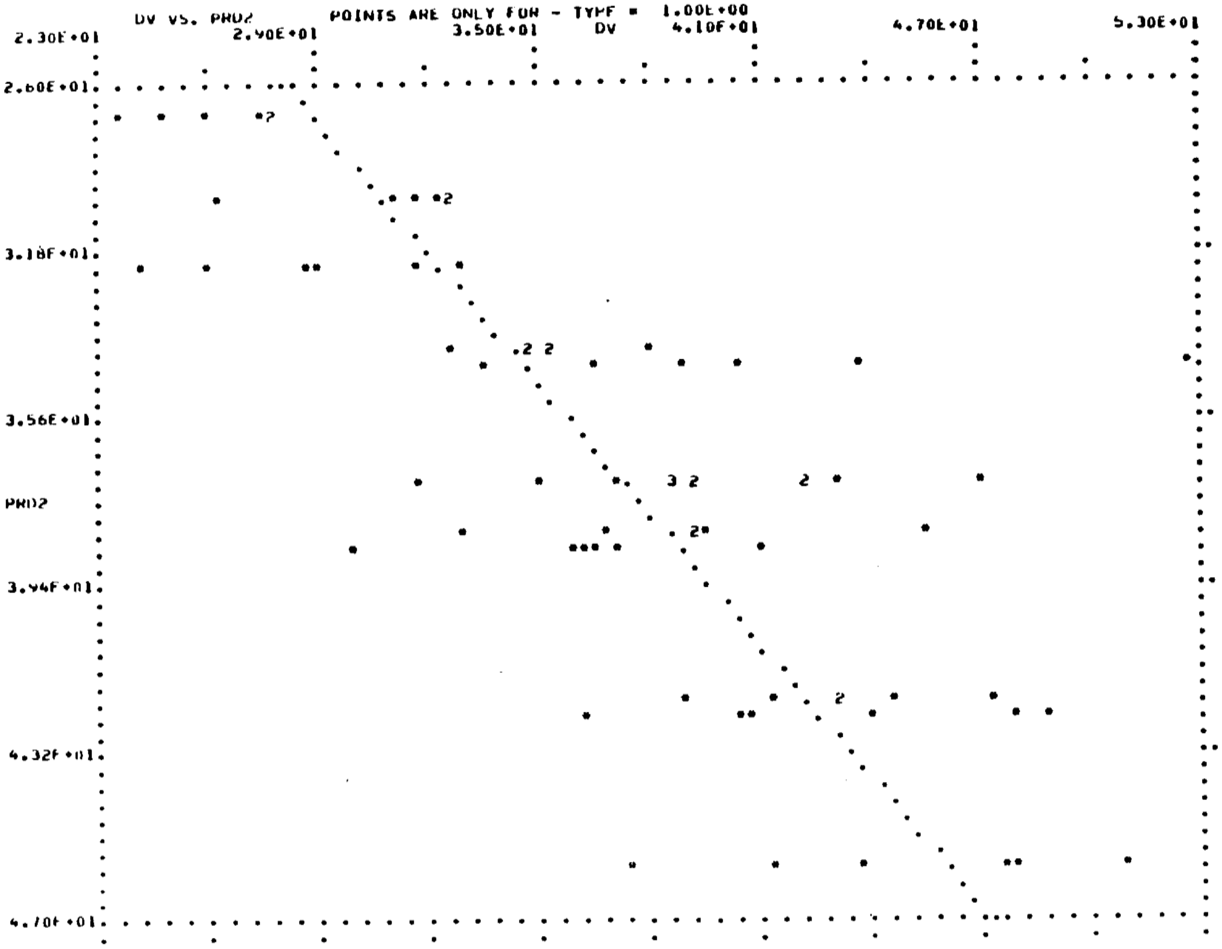
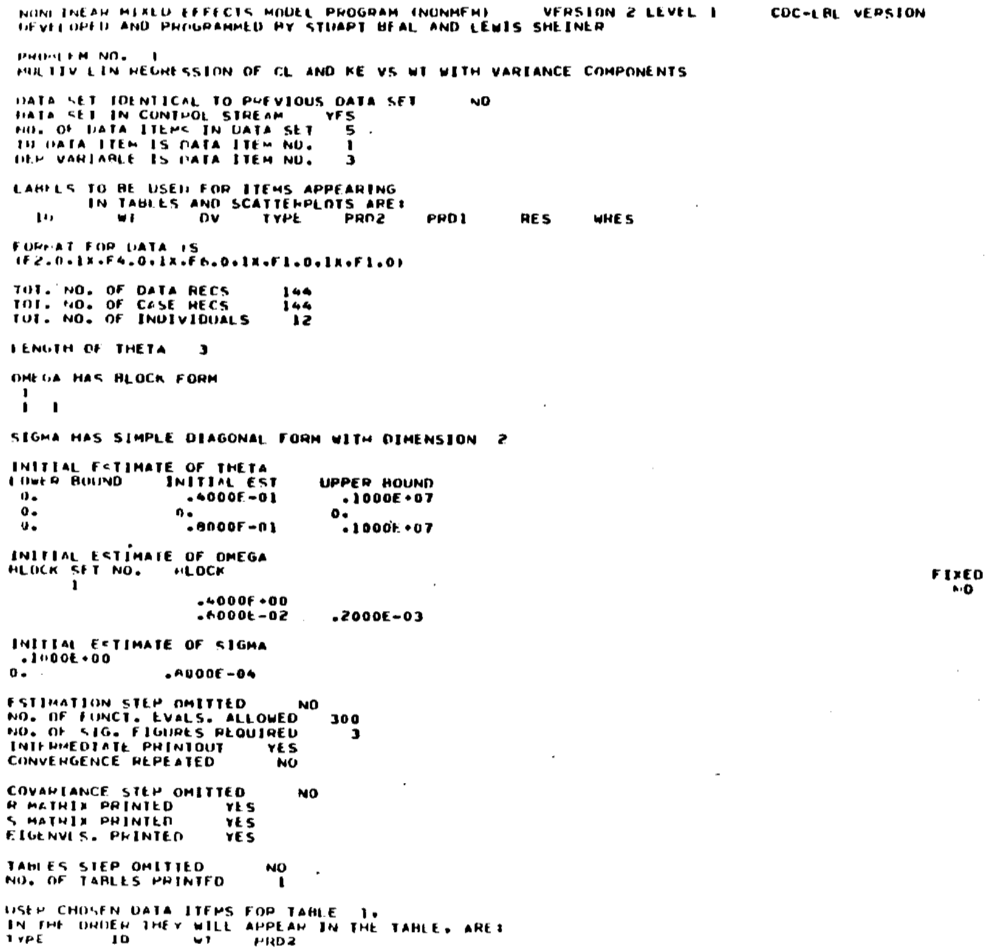


Figure 12

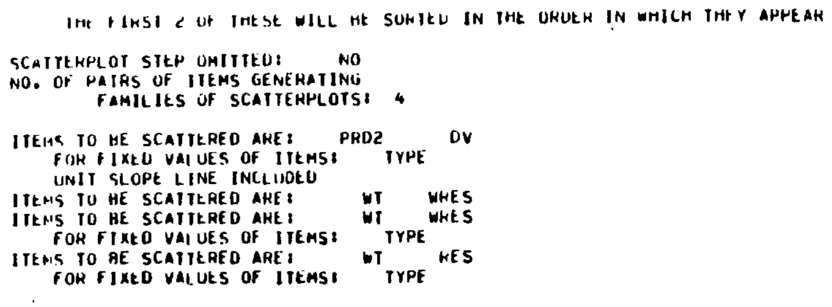
Figure 13a



```
NONLINEAR MIXED EFFECTS MODEL PROGRAM (NONMEM)      VERSION 2 LEVEL 1      CDC-LBL VERSION
DEVELOPED AND PROGRAMMED BY STUART BEAL AND LEWIS SHEINER

PROBLEM NO.   1
MULTIV LIN REGRESSION OF CL AND KE VS WT WITH VARIANCE COMPONENTS

DATA SET IDENTICAL TO PREVIOUS DATA SET      NO
DATA SET IN CONTROL STREAM     YES
NO. OF DATA ITEMS IN DATA SET   5
ID DATA ITEM IS DATA ITEM NO.   1
DEP VARIABLE IS DATA ITEM NO.   3

LABELS TO BE USED FOR ITEMS APPEARING
       IN TABLES AND SCATTERPLOTS ARE:
  ID       WT       DV     TYPE     PRD2     PRD1      RES     WRES

FORMAT FOR DATA IS
(F2.0,1X,F4.0,1X,F6.0,1X,F1.0,1X,F1.0)

TOT. NO. OF DATA RECS      144
TOT. NO. OF CASE RECS      144
TOT. NO. OF INDIVIDUALS     12

LENGTH OF THETA    3

OMEGA HAS BLOCK FORM
 1
 1  1

SIGMA HAS SIMPLE DIAGONAL FORM WITH DIMENSION  2

INITIAL ESTIMATE OF THETA
LOWER BOUND    INITIAL EST    UPPER BOUND
 0.              .4000E-01      .1000E+07
 0.             0.             0.
 0.              .8000E-01      .1000E+07

INITIAL ESTIMATE OF OMEGA
BLOCK SET NO.    BLOCK                                                           FIXED
       1                                                                          NO
                 .4000E+00
                 .6000E-02    .2000E-03

INITIAL ESTIMATE OF SIGMA
 .1000E+00
0.              .8000E-04

ESTIMATION STEP OMITTED      NO
NO. OF FUNCT. EVALS. ALLOWED    300
NO. OF SIG. FIGURES REQUIRED      3
INTERMEDIATE PRINTOUT     YES
CONVERGENCE REPEATED       NO

COVARIANCE STEP OMITTED      NO
R MATRIX PRINTED      YES
S MATRIX PRINTED      YES
EIGENVLS. PRINTED     YES

TABLES STEP OMITTED      NO
NO. OF TABLES PRINTED     1

USER CHOSEN DATA ITEMS FOR TABLE  1,
IN THE ORDER THEY WILL APPEAR IN THE TABLE, ARE:
TYPE       ID       WT     PRD2
```

Figure 13a

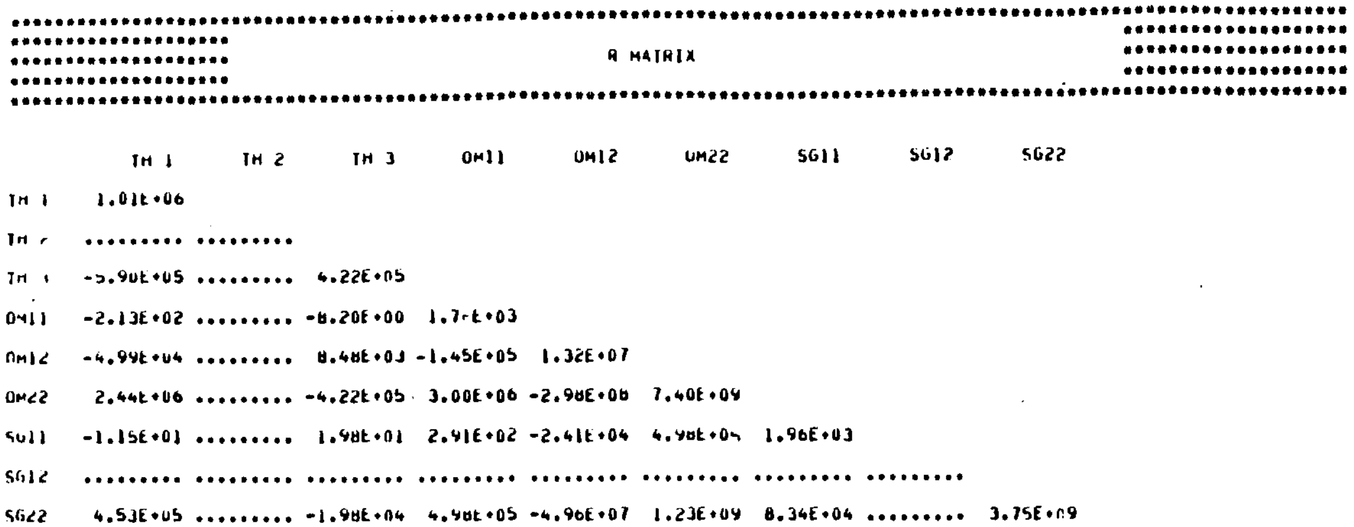
Figure 13b



```
      THE FIRST 2 OF THESE WILL BE SORTED IN THE ORDER IN WHICH THEY APPEAR

 SCATTERPLOT STEP OMITTED:      NO
 NO. OF PAIRS OF ITEMS GENERATING
        FAMILIES OF SCATTERPLOTS:  4

 ITEMS TO BE SCATTERED ARE:     PRD2       DV
     FOR FIXED VALUES OF ITEMS:     TYPE
     UNIT SLOPE LINE INCLUDED
 ITEMS TO BE SCATTERED ARE:       WT     WRES
 ITEMS TO BE SCATTERED ARE:       WT     WRES
     FOR FIXED VALUES OF ITEMS:     TYPE
 ITEMS TO BE SCATTERED ARE:       WT      RES
     FOR FIXED VALUES OF ITEMS:     TYPE
```

Figure 13b

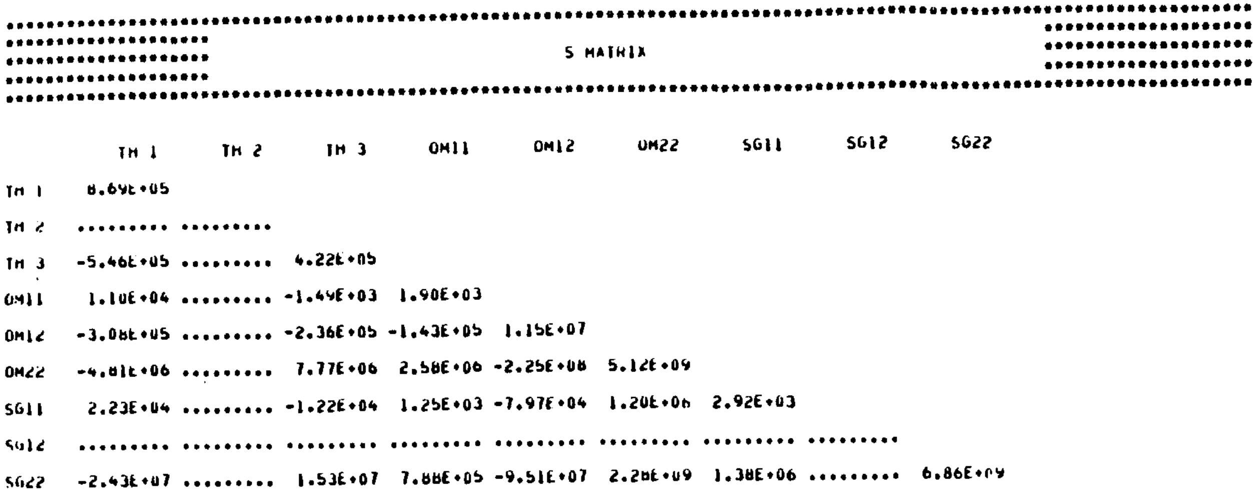
Figure 14



```
*******************************************************************************************************************
********************                                                                          ********************
********************                              R MATRIX                                    ********************
********************                                                                          ********************
*******************************************************************************************************************

            TH 1      TH 2      TH 3      OM11      OM12      OM22      SG11      SG12      SG22

 TH 1     1.01E+06

 TH 2    ......... .........

 TH 3    -5.90E+05 .........  4.22E+05

 OM11    -2.13E+02 ......... -8.20E+00  1.7-E+03

 OM12    -4.99E+04 .........  8.48E+03 -1.45E+05  1.32E+07

 OM22     2.44E+06 ......... -4.22E+05  3.00E+06 -2.98E+06  7.40E+09

 SG11    -1.15E+01 .........  1.98E+01  2.91E+02 -2.41E+04  4.98E+04  1.96E+03

 SG12    ......... ......... ......... ......... ......... ......... ......... .........

 SG22     4.53E+05 ......... -1.98E+04  4.98E+05 -4.96E+07  1.23E+09  8.34E+04 .........  3.75E+09
```

Figure 14

Figure 15

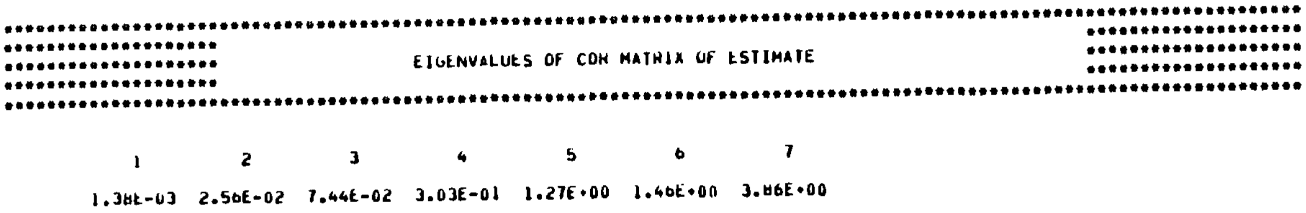


```
 ************************************************************************************************************************
 ********************                                                                                ********************
 ********************                                 S MATRIX                                       ********************
 ********************                                                                                ********************
 ************************************************************************************************************************


            TH 1      TH 2      TH 3      OM11      OM12      OM22      SG11      SG12      SG22

 TH 1     8.69E+05

 TH 2    ......... .........

 TH 3    -5.46E+05 .........  4.22E+05

 OM11     1.10E+04 ......... -1.49E+03  1.90E+03

 OM12    -3.08E+05 ......... -2.36E+05 -1.43E+05  1.15E+07

 OM22    -4.81E+06 .........  7.77E+06  2.58E+06 -2.25E+08  5.12E+09

 SG11     2.23E+04 ......... -1.22E+04  1.25E+03 -7.97E+04  1.20E+06  2.92E+03

 SG12    ......... ......... ......... ......... ......... ......... ......... .........

 SG22    -2.43E+07 .........  1.53E+07  7.88E+05 -9.51E+07  2.28E+09  1.38E+06 .........  6.86E+09
```

Figure 15

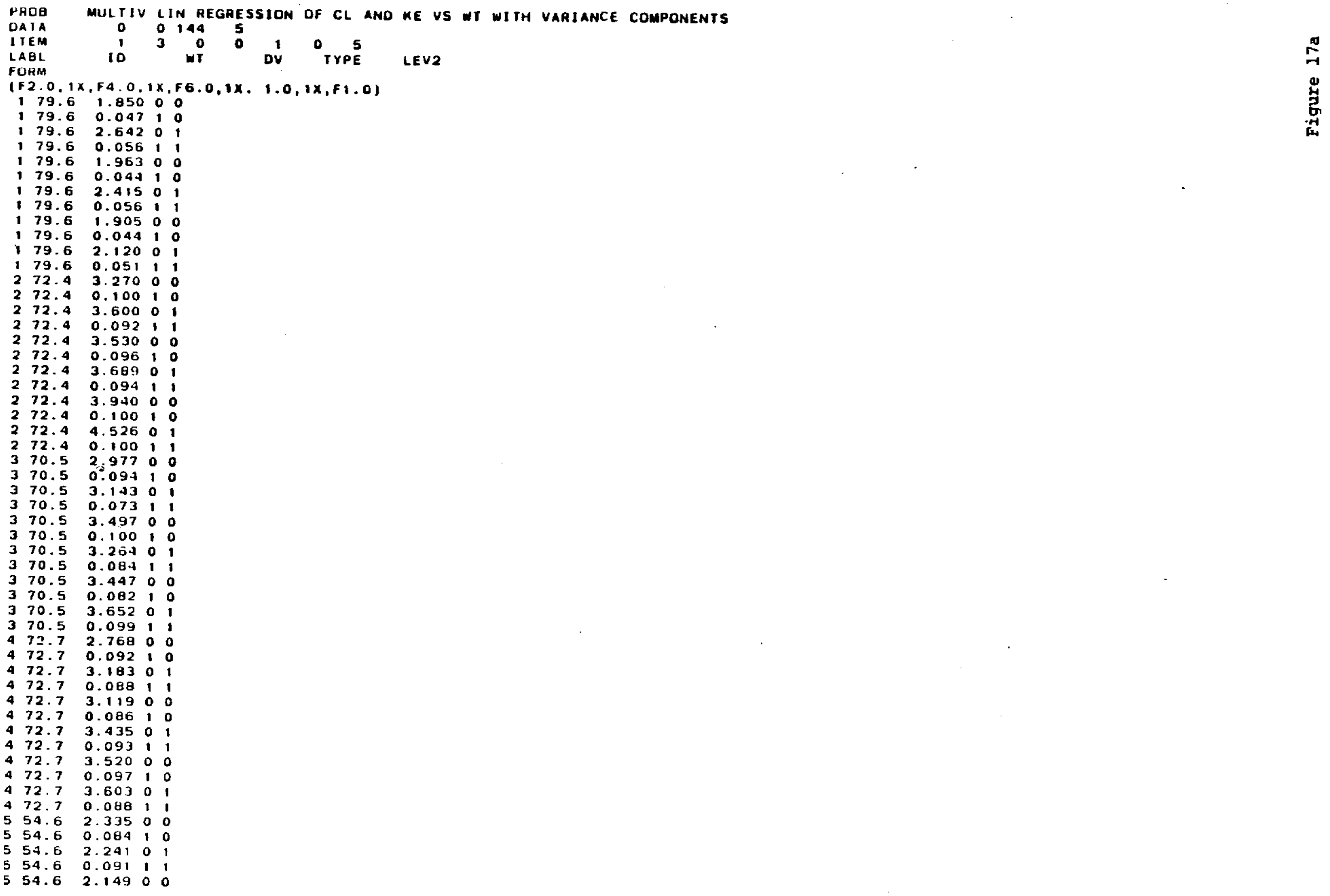
Figure 16

Figure 16



```
************************************************************************************************************************************
********************                                                                                         ********************
********************                   EIGENVALUES OF COR MATRIX OF ESTIMATE                                 ********************
********************                                                                                         ********************
************************************************************************************************************************************

            1          2          3          4          5          6          7

         1.38E-03   2.56E-02   7.44E-02   3.03E-01   1.27E+00   1.46E+00   3.86E+00
```

Figure 17a



```
PROB     MULTIV LIN REGRESSION OF CL AND KE VS WT WITH VARIANCE COMPONENTS
DATA        0    0 144    5
ITEM        1    3    0    0    1    0    5
LABL       ID         WT         DV     TYPE      LEV2
FORM
(F2.0,1X,F4.0,1X,F6.0,1X. 1.0,1X,F1.0)
 1 79.6  1.850 0 0
 1 79.6  0.047 1 0
 1 79.6  2.642 0 1
 1 79.6  0.056 1 1
 1 79.6  1.963 0 0
 1 79.6  0.044 1 0
 1 79.6  2.415 0 1
 1 79.6  0.056 1 1
 1 79.6  1.905 0 0
 1 79.6  0.044 1 0
 1 79.6  2.120 0 1
 1 79.6  0.051 1 1
 2 72.4  3.270 0 0
 2 72.4  0.100 1 0
 2 72.4  3.600 0 1
 2 72.4  0.092 1 1
 2 72.4  3.530 0 0
 2 72.4  0.096 1 0
 2 72.4  3.689 0 1
 2 72.4  0.094 1 1
 2 72.4  3.940 0 0
 2 72.4  0.100 1 0
 2 72.4  4.526 0 1
 2 72.4  0.100 1 1
 3 70.5  2.977 0 0
 3 70.5  0.094 1 0
 3 70.5  3.143 0 1
 3 70.5  0.073 1 1
 3 70.5  3.497 0 0
 3 70.5  0.100 1 0
 3 70.5  3.264 0 1
 3 70.5  0.084 1 1
 3 70.5  3.447 0 0
 3 70.5  0.082 1 0
 3 70.5  3.652 0 1
 3 70.5  0.099 1 1
 4 72.7  2.768 0 0
 4 72.7  0.092 1 0
 4 72.7  3.183 0 1
 4 72.7  0.088 1 1
 4 72.7  3.119 0 0
 4 72.7  0.086 1 0
 4 72.7  3.435 0 1
 4 72.7  0.093 1 1
 4 72.7  3.520 0 0
 4 72.7  0.097 1 0
 4 72.7  3.603 0 1
 4 72.7  0.088 1 1
 5 54.6  2.335 0 0
 5 54.6  0.084 1 0
 5 54.6  2.241 0 1
 5 54.6  0.091 1 1
 5 54.6  2.149 0 0
```

Figure 17a

Figure 17b

```
 5 54.6   0.091 1 0
 5 54.6   2.381 0 1
 5 54.6   0.087 1 1
 5 54.6   2.184 0 0
 5 54.6   0.084 1 0
 5 54.6   1.805 0 1
 5 54.6   0.065 1 1
 6 80.0   3.885 0 0
 6 80.0   0.088 1 0
 6 80.0   3.079 0 1
 6 80.0   0.076 1 1
 6 80.0   3.600 0 0
 6 80.0   0.074 1 0
 6 80.0   3.963 0 1
 6 80.0   0.098 1 1
 6 80.0   3.598 0 0
 6 80.0   0.075 1 0
 6 80.0   3.415 0 1
 6 80.0   0.095 1 1
 7 64.6   3.175 0 0
 7 64.6   0.090 1 0
 7 64.6   3.260 0 1
 7 64.6   0.100 1 1
 7 64.6   3.590 0 0
 7 64.6   0.103 1 0
 7 64.6   3.154 0 1
 7 64.6   0.089 1 1
 7 64.6   3.616 0 0
 7 64.6   0.095 1 0
 7 64.6   3.027 0 1
 7 64.6   0.087 1 1
 8 70.5   3.140 0 0
 8 70.5   0.081 1 0
 8 70.5   3.310 0 1
 8 70.5   0.086 1 1
 8 70.5   3.426 0 0
 8 70.5   0.087 1 0
 8 70.5   3.445 0 1
 8 70.5   0.073 1 1
 8 70.5   3.237 0 0
 8 70.5   0.077 1 0
 8 70.5   3.279 0 1
 8 70.5   0.083 1 1
 9 86.4   3.247 0 0
 9 86.4   0.078 1 0
 9 86.4   2.628 0 1
 9 86.4   0.055 1 1
 9 86.4   3.296 0 0
 9 86.4   0.088 1 0
 9 86.4   3.380 0 1
 9 86.4   0.066 1 1
 9 86.4   3.621 0 0
 9 86.4   0.076 1 0
 9 86.4   3.240 0 1
 9 86.4   0.074 1 1
10 58.2   1.889 0 0
10 58.2   0.072 1 0
10 58.2   2.800 0 1
10 58.2   0.090 1 1
```

Figure 17b

Figure 17c

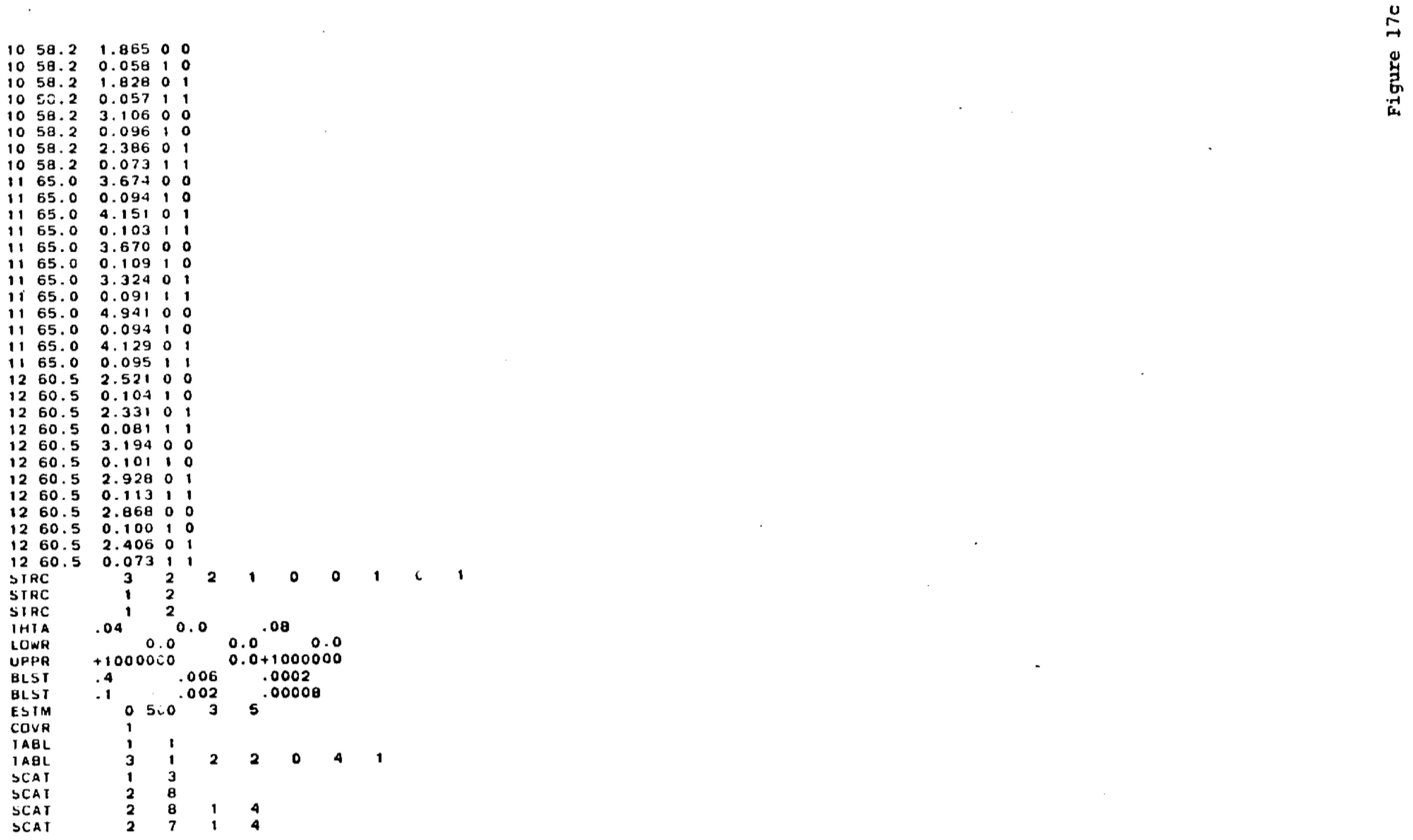
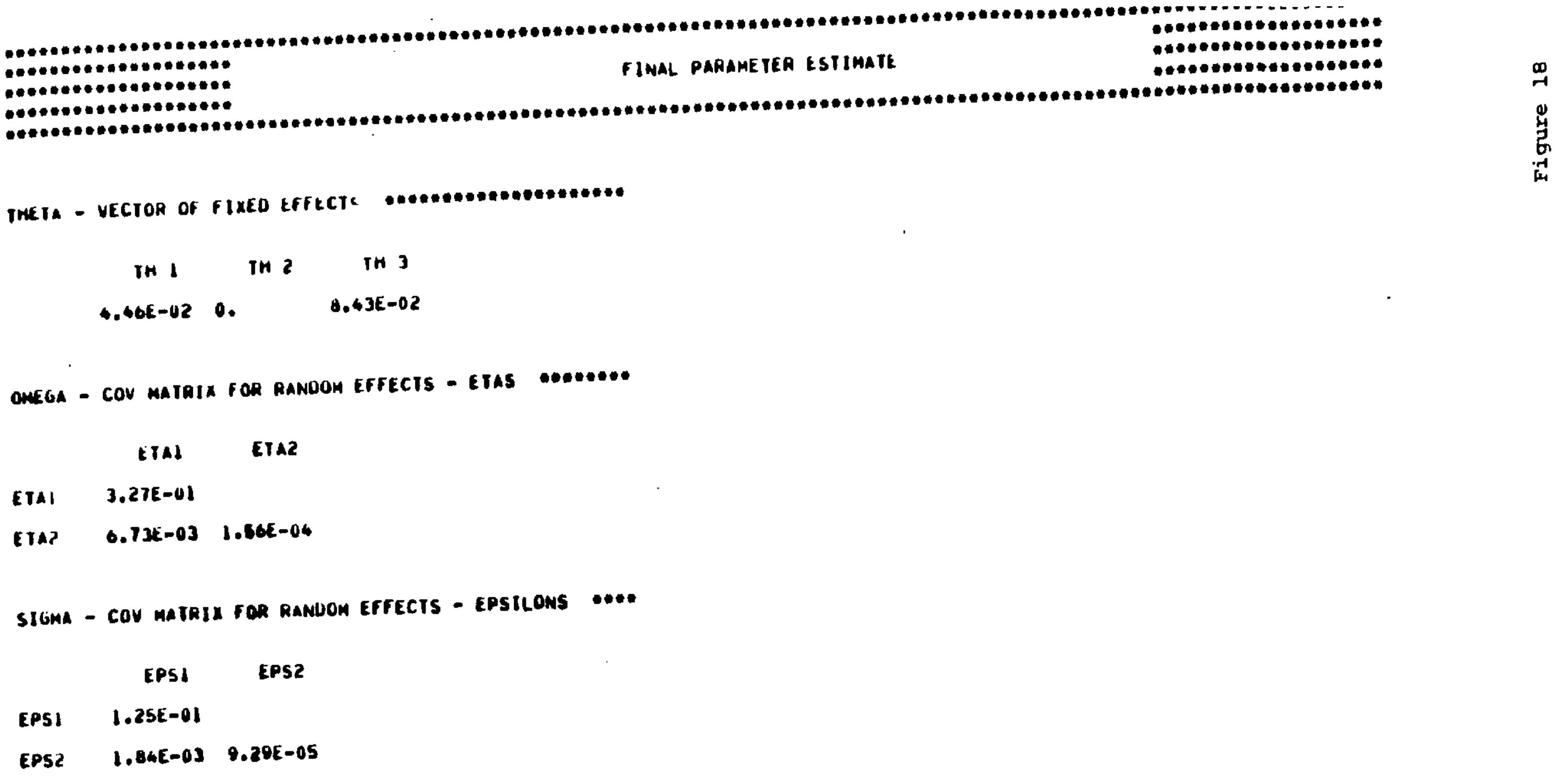


Figure 17c

```
10 58.2   1.865 0 0
10 58.2   0.058 1 0
10 58.2   1.828 0 1
10 58.2   0.057 1 1
10 58.2   3.106 0 0
10 58.2   0.096 1 0
10 58.2   2.386 0 1
10 58.2   0.073 1 1
11 65.0   3.674 0 0
11 65.0   0.094 1 0
11 65.0   4.151 0 1
11 65.0   0.103 1 1
11 65.0   3.670 0 0
11 65.0   0.109 1 0
11 65.0   3.324 0 1
11 65.0   0.091 1 1
11 65.0   4.941 0 0
11 65.0   0.094 1 0
11 65.0   4.129 0 1
11 65.0   0.095 1 1
12 60.5   2.521 0 0
12 60.5   0.104 1 0
12 60.5   2.331 0 1
12 60.5   0.081 1 1
12 60.5   3.194 0 0
12 60.5   0.101 1 0
12 60.5   2.928 0 1
12 60.5   0.113 1 1
12 60.5   2.868 0 0
12 60.5   0.100 1 0
12 60.5   2.406 0 1
12 60.5   0.073 1 1
STRC         3    2    2    1    0    0    1    0    1
STRC         1    2
STRC         1    2
THTA     .04       0.0       .08
LOWR           0.0       0.0       0.0
UPPR     +1000000        0.0+1000000
BLST     .4        .006      .0002
BLST     .1        .002      .00008
ESTM         0  500    3    5
COVR         1
TABL         1    1
TABL         3    1    2    2    0    4    1
SCAT         1    3
SCAT         2    8
SCAT         2    8    1    4
SCAT         2    7    1    4
```

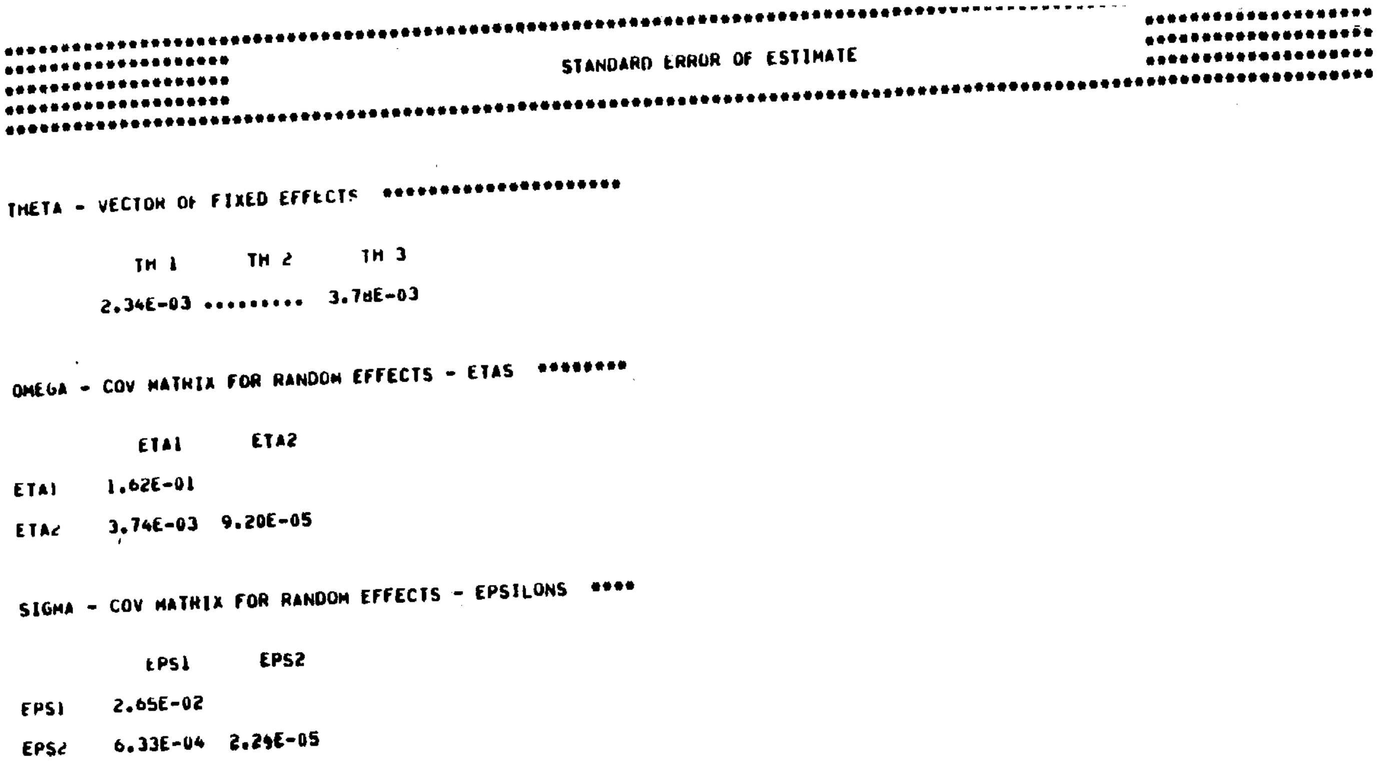
Figure 18



```
 *******************************************************************************************************************************
 **********************                                                                                     **********************
 **********************                                FINAL PARAMETER ESTIMATE                             **********************
 **********************                                                                                     **********************
 *******************************************************************************************************************************



 THETA - VECTOR OF FIXED EFFECTS  ***********************


            TH 1        TH 2        TH 3

         4.46E-02  0.          8.43E-02



 OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********


            ETA1        ETA2

 ETA1     3.27E-01

 ETA2     6.73E-03  1.66E-04



 SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****


            EPS1        EPS2

 EPS1     1.25E-01

 EPS2     1.84E-03  9.29E-05
```

Figure 18

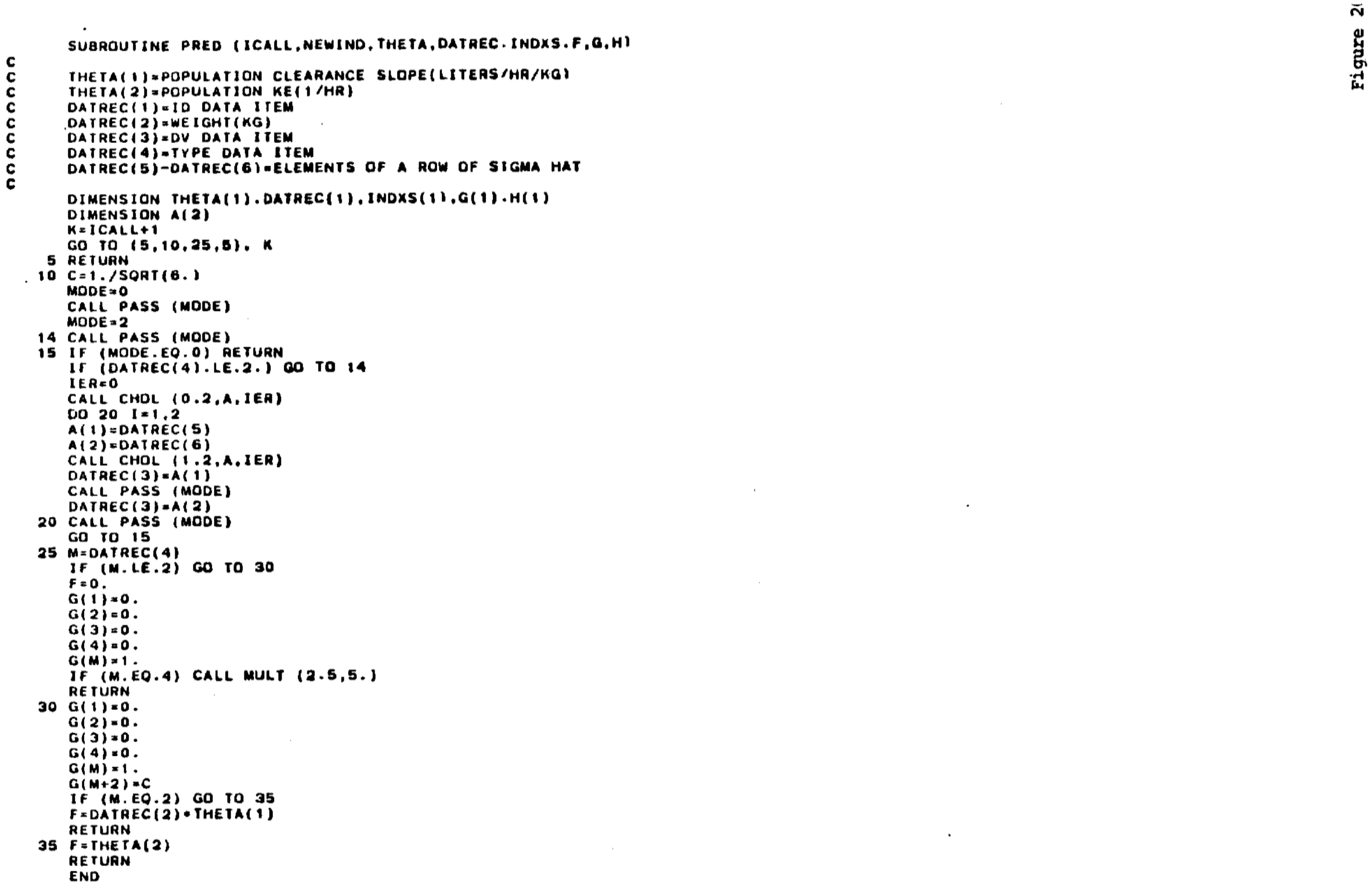
Figure 19



```
************************************************************************************************************************************
********************                                                                                          ********************
********************                          STANDARD ERROR OF ESTIMATE                                      ********************
********************                                                                                          ********************
************************************************************************************************************************************


THETA - VECTOR OF FIXED EFFECTS  ********************


            TH 1      TH 2      TH 3

         2.34E-03 *********  3.78E-03


OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********


            ETA1      ETA2

ETA1     1.62E-01

ETA2     3.74E-03  9.20E-05


SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****


            EPS1      EPS2

EPS1     2.65E-02

EPS2     6.33E-04  2.29E-05
```

Figure 19

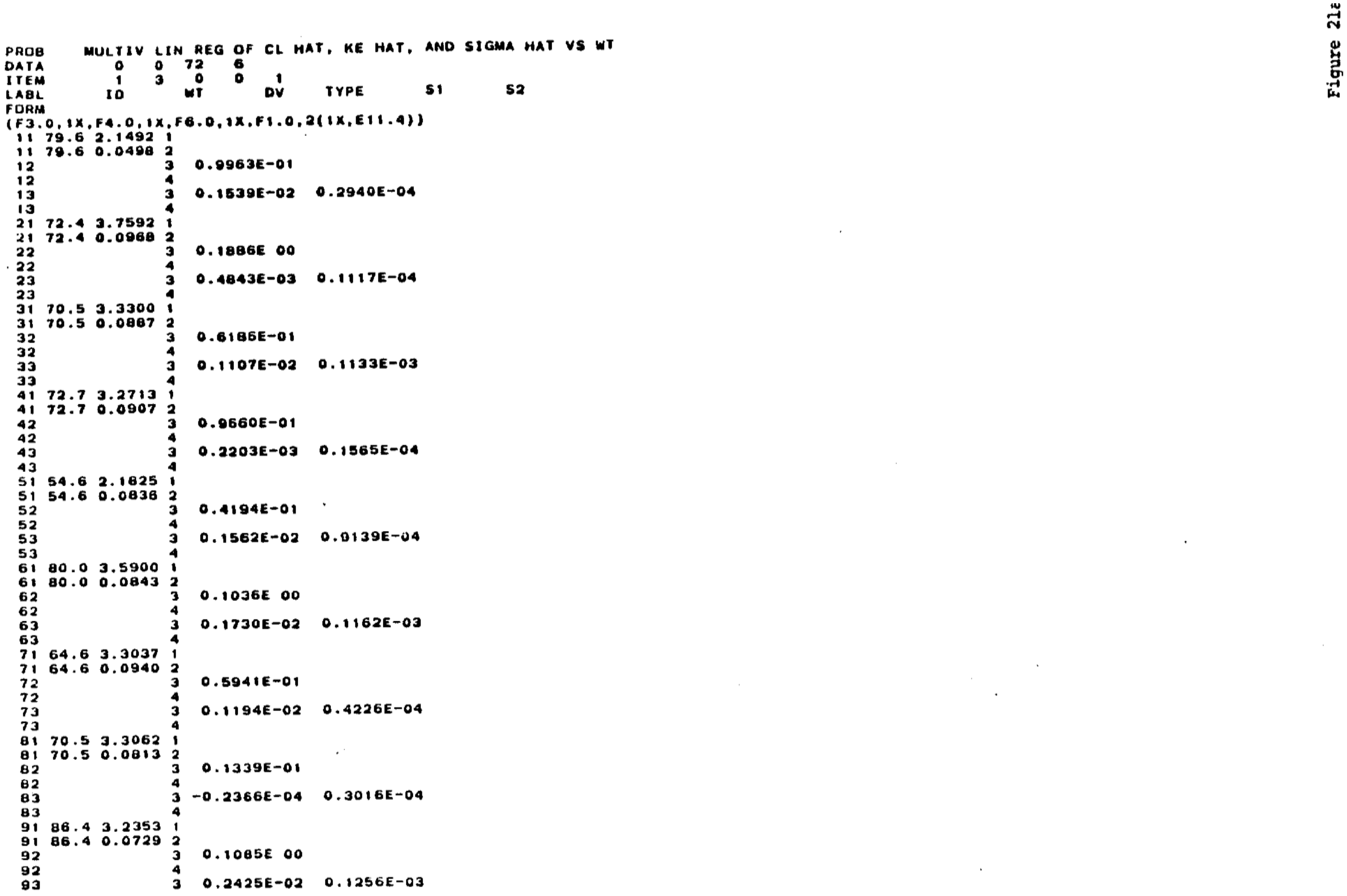
Figure 20



```
      SUBROUTINE PRED (ICALL,NEWIND,THETA,DATREC.INDXS.F,G,H)
C
C     THETA(1)=POPULATION CLEARANCE SLOPE(LITERS/HR/KG)
C     THETA(2)=POPULATION KE(1/HR)
C     DATREC(1)=ID DATA ITEM
C     DATREC(2)=WEIGHT(KG)
C     DATREC(3)=DV DATA ITEM
C     DATREC(4)=TYPE DATA ITEM
C     DATREC(5)-DATREC(6)=ELEMENTS OF A ROW OF SIGMA HAT
C
      DIMENSION THETA(1).DATREC(1),INDXS(1),G(1).H(1)
      DIMENSION A(2)
      K=ICALL+1
      GO TO (5,10,25,5). K
    5 RETURN
   10 C=1./SQRT(6.)
      MODE=0
      CALL PASS (MODE)
      MODE=2
   14 CALL PASS (MODE)
   15 IF (MODE.EQ.0) RETURN
      IF (DATREC(4).LE.2.) GO TO 14
      IER=0
      CALL CHOL (0.2,A,IER)
      DO 20 I=1,2
      A(1)=DATREC(5)
      A(2)=DATREC(6)
      CALL CHOL (1.2,A,IER)
      DATREC(3)=A(1)
      CALL PASS (MODE)
      DATREC(3)=A(2)
   20 CALL PASS (MODE)
      GO TO 15
   25 M=DATREC(4)
      IF (M.LE.2) GO TO 30
      F=0.
      G(1)=0.
      G(2)=0.
      G(3)=0.
      G(4)=0.
      G(M)=1.
      IF (M.EQ.4) CALL MULT (2.5,5.)
      RETURN
   30 G(1)=0.
      G(2)=0.
      G(3)=0.
      G(4)=0.
      G(M)=1.
      G(M+2)=C
      IF (M.EQ.2) GO TO 35
      F=DATREC(2)*THETA(1)
      RETURN
   35 F=THETA(2)
      RETURN
      END
```

Figure 2

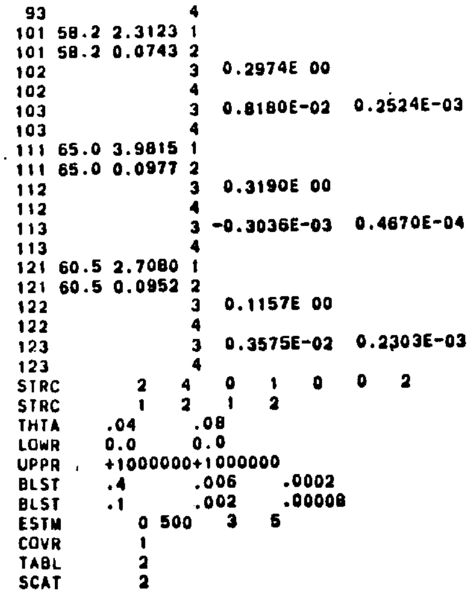
Figure 21a



```
PROB     MULTIV LIN REG OF CL HAT, KE HAT, AND SIGMA HAT VS WT
DATA        0    0   72    6
ITEM        1    3    0    0    1
LABL       ID        WT        DV     TYPE       S1        S2
FORM
(F3.0,1X,F4.0,1X,F6.0,1X,F1.0,2(1X,E11.4))
 11 79.6 2.1492 1
 11 79.6 0.0498 2
 12             3  0.9963E-01
 12             4
 13             3  0.1539E-02  0.2940E-04
 13             4
 21 72.4 3.7592 1
 21 72.4 0.0968 2
 22             3  0.1886E 00
 22             4
 23             3  0.4843E-03  0.1117E-04
 23             4
 31 70.5 3.3300 1
 31 70.5 0.0887 2
 32             3  0.6185E-01
 32             4
 33             3  0.1107E-02  0.1133E-03
 33             4
 41 72.7 3.2713 1
 41 72.7 0.0907 2
 42             3  0.9660E-01
 42             4
 43             3  0.2203E-03  0.1565E-04
 43             4
 51 54.6 2.1825 1
 51 54.6 0.0836 2
 52             3  0.4194E-01
 52             4
 53             3  0.1562E-02  0.0139E-04
 53             4
 61 80.0 3.5900 1
 61 80.0 0.0843 2
 62             3  0.1036E 00
 62             4
 63             3  0.1730E-02  0.1162E-03
 63             4
 71 64.6 3.3037 1
 71 64.6 0.0940 2
 72             3  0.5941E-01
 72             4
 73             3  0.1194E-02  0.4226E-04
 73             4
 81 70.5 3.3062 1
 81 70.5 0.0813 2
 82             3  0.1339E-01
 82             4
 83             3 -0.2366E-04  0.3016E-04
 83             4
 91 86.4 3.2353 1
 91 86.4 0.0729 2
 92             3  0.1085E 00
 92             4
 93             3  0.2425E-02  0.1256E-03
```

Figure 21a

Figure 21b



```
 93             4
101 58.2 2.3123 1
101 58.2 0.0743 2
102             3  0.2974E 00
102             4
103             3  0.8180E-02  0.2524E-03
103             4
111 65.0 3.9815 1
111 65.0 0.0977 2
112             3  0.3190E 00
112             4
113             3 -0.3036E-03  0.4670E-04
113             4
121 60.5 2.7080 1
121 60.5 0.0952 2
122             3  0.1157E 00
122             4
123             3  0.3575E-02  0.2303E-03
123             4
STRC       2   4   0   1   0   0   2
STRC       1   2   1   2
THTA    .04     .08
LOWR    0.0     0.0
UPPR    +1000000+1000000
BLST    .4      .006    .0002
BLST    .1      .002    .00008
ESTM       0 500   3   5
COVR       1
TABL       2
SCAT       2
```

Figure 21b

E. Multiple Problems

E.1 Introduction

The possibility for multiple problem specifications to be processed in a single NONMEM run is discussed in this section. A single problem specification is just a sequence of control records (possibly with embedded FORTRAN records that generate a data set) that completely specify one problem. It consists of a single problem record, one or more (contiguous) Data Set Specification records, one or more (contiguous) Model Specification records, and one or more (contiguous) Task Specification records. A multiple problem specification (MPS) is simply the concatenation of two or more single problem specifications. Each of the single problem specifications comprising a multiple problem specification is also called an individual problem specification (IPS). If one individual problem specification immediately follows another in the MPS, these two individual problem specifications are called contiguous. If in a sequence of individual problem specifications, $A_{1,} A_2, ..., A_n$ for each $i < n$, A_i and A_{i+1} are contiguous, the sequence is called contiguous. Essentially, all that is necessary to process a multiple problem specification is to have the control stream (i.e. the input file containing the control records) consist of the MPS. Each IPS is processed in the order it appears in the MPS. Simple as all this may be, there still are aspects of the MPS that must be described, and there are certain features that a user may find advantageous. These matters are discussed in the subsequent sections.

E.2 Sharing a Data Set

In a MPS situation, the FORTRAN records generating data sets may be placed in a sequential input file separate from the control stream. If the first n FORTRAN records of this file generate a data set to be used with two or more individual problem specifications, then the user will want to rewind the FORTRAN unit containing this file. As mentioned in section 3.5 of NONMEM Users Guide, Part I, this can be done by placing a 1 in field 2 of the DATA record of any IPS with which the rewind is to occur.

An alternative way to use the same data set with two or more individual problem specifications is the following way. Suppose that a particular data set is used with one individual problem specification and that this data set is to be used with one or more succeeding individual problem specifications that, together with the first one, form a contiguous sequence of individual problem specifications. The data set is established with the first individual problem specification, and it need not be reestablished with the other problem specifications of the sequence. It has a representation internal to NONMEM that may be used with these other problem specifications. To refer NONMEM to this representation, in each of these other problem specifications it is only necessary to place a -1 in field 1 of the DATA records of each of them. In this case a FORMAT record need not and should not appear, and, of course, embedded data should not appear also, in any of the problem specifications of the sequence except the first one. In addition, all values placed in other fields of a DATA record with a -1 in field 1 are ignored. The number of data items per data record in each problem specification of the sequence is the same as the number of data items per record in the first one. (However, it is explained below how data items themselves can actually be made to vary between problems.) The complete format of the DATA record is given in Table E.2.i

Table E.2.i DATA record format ***(See Introduction to NONMEM VI March 2008 for current format.)***

Field No.	Value	Function
1	-1 0 or blank between 20 & 99	Data set same as that for previous problem specification Data file embedded in the control stream FORTRAN unit number for data file
If the value is -1, the subsequent fields may be ignored.		
2	0 or blank 1	FORTRAN unit not to be rewound FORTRAN unit to be rewound
3	0 or blank positive integer	FINISH record used number of data records
4	between 1 & 20	number of data items per data record

The reader will recall that a data set may be modified with the use of the NONMEM utility PASS at either PRED initialization or finalization (see section D.2). At such a time, PRED can also write the modified data set onto a peripheral file so that this file may be read as the input data file in a subsequent problem in the same NONMEM run. Use of field 1 as just described will accomplish the same end when the subsequent problem immediately follows the current problem in the same run. The data set, as it exists at the end of PRED finalization, is "passed on" to the next problem. All data items may be modified by PRED at PRED initialization and/or finalization of the current problem, except for the ID and MDV data items. The ID and MDV data items may not be modified at any time. However, the indices of the DV, MDV and L2 data items may vary between problem specifications referring to the "same" data set. So, in particular, the MDV data items may differ between problem specifications referring to the "same" data set.

E.3 Model Specification Files

(See Help Items in Guide VIII, March 2008 for current information.)

Recall that the Model Specification File is physically one logical record of a sequential data set. Different individual problem specifications may specify different Model Specification Files be output onto the same data set (in which case in the data definition statement of the data set the disposition should be MOD.) Likewise, different individual problem specifications may specify different Model Specification Files be input from the same data set. If the Model Specification Files to be input, $A_{1,} A_2 ,..., A_n$, appear in the data set in the indicated order (with possibly other Model Specification Files between them), and this order is the one in which they are to be input, then simply specifying their identifiers on the FIND records allows them to be retrieved. However, if A_1 , is to be input after A_2 , say, then the FORTRAN unit containing the data set should be rewound after A_2 is input and before A_1 is input. The unit will be rewound before the Model Specification File is input if a 1 is placed in field 3 of the FIND record for that Model Specification File. Table E.3.i gives the format for the FIND record through this third field.

Table E.3.i FIND record ***(See Introduction to NONMEM VI March 2008 for current format.)***

Field No.	**Value**	**Function**
1	between 1 & 9999	identifier for (existing) Model Specification File
2	between 20 & 99	FORTRAN unit number for (existing) Model Specification File
3	3 or blank 1	FORTRAN unit not to be rewound FORTRAN unit to be rewound

E.4 Embedding Files Into the Control Stream

The control stream is a sequential input file comprised essentially of the control records. However, sequential files of other types of records may be embedded in the control stream. For example, an input data file may be embedded. In addition though, two user supplied routines, PRED and CRIT, may require some sequential input files, and any of these files may also be embedded. Call the files for PRED to be embedded, PRED embedded files, and call the files for CRIT to be embedded, CRIT embedded files. Suppose F is a file to be embedded. To embed F in the control stream means to insert the records of F, altogether and in sequence, after either a control record or the last record of another embedded file. This means that the records of F, like the control records, must be 80 character records. If one file, F_2, is inserted farther down the control stream than another file, F_1 , then F_2 is said to be below F_1. As explained below, files must be embedded only at particular places in the control stream.

An input data file for a given problem should be embedded after the F0RMAT record for that problem.

The control stream is sequential and is read once only. Therefore, each PRED or CRIT embedded file, F, must be read entirely before any record of any other such file below F is read. Also therefore, F can be read once only.

It is now explained where the PRED and CRIT embedded data sets should be embedded in the control stream. Consider first a given problem, but not the first one. The situation for the first problem is a little more complicated and is described below. A11 CRIT embedded files for the problem (i.e. files read by CRIT when ICALL is 1 for the problem) must be below all PRED embedded files for the problem (i.e. files read by PRED when ICALL is either 1 or 3 for the problem). The first of the PRED and CRIT embedded files for the problem should be embedded after the last SCATTERPLOT record of the problem specification. There must be no control records between any of the PRED or CRIT embedded files for the problem. These PRED and CRIT embedded files are regarded as part of the problem specification.

For the purpose of this description then, all the PRED embedded files for the kth problem may be regarded as a single PRED embedded file, P(k). This file could be empty. Similarly, all CRIT embedded files for the kth problem may be regarded as a single CRIT embedded file, C(k). This file, too, could be empty. From the above it follows that when $k \geq 2$, $C(k)$ should occur below $P(k)$ without any intervening control records, and that $P(k)$ should be embedded after the last SCATTERPLOT record of the problem specification.

Now similarly, the PRED embedded files for NONMEM initialization (i.e. files read by PRED when ICALL is 0) should be regarded as a single PRED embedded file, P(0). Also, the CRIT embedded files for NONMEM initialization (i.e. files read by CRIT when ICALL is 0) should be regarded as a single CRIT embedded file, C(0). Then C(1) should occur below C(0), C(0) below P(1), P(1) below P(0), without any intervening control records, and P(0) should be embedded after the last SCATTERPLOT record of the first problem specification. The foregoing is summarized in Table E.4.i.

Table E.4.i Control Stream

```
PROBLEM record
…
FORMAT record
            <---------[input data set]
STRUCTURE record

…
SCATTERPLOT record

            <---------[P(0)]
            <---------[P(1)]
            <---------[C(0)]
            <---------[C(1)]
PROBLEM record
…
FORMAT record
            <---------[input data set]
STRUCTURE record

…
SCATTERPLOT record

            <---------[P(2)]
            <---------[C(2)]
PROBLEM record
…
```

E.5 Embedding Files into the Print Stream

NONMEM printed output is placed in a print file, PF. Print files generated by PRED or CRIT may be embedded in PF. Call the files from PRED to be embedded, PRED embedded files, and those from CRIT to be embedded, CRIT embedded files. Suppose F is a print file to be embedded, to embed F in PF means to input the records of F altogether after either a page of NONMEM print records or the last page of another embedded print file. (It is the responsibility of PRED (or CRIT) to insert a print control character that begins a new page of a PRED (or CRIT) embedded file when a new page is to begin.) If one file, F_2 is inserted further down the PF than another file F_1, then F_2 is said to be below F_1.

PF is, of course, sequential. Therefore, each PRED or CRIT embedded file, F, should be written entirely before any record of any other such file below F is written.

It is now explained where the PRED and CRIT embedded files are embedded in PP. Consider first a given problem, but not the first one. The situation for the first problem is a little more complicated, and is described below. There are two types of PRED embedded files for the problem. The first type is generated at problem initialization (i.e. when ICALL is 1), and the second type is generated at problem finalization (i.e. when ICALL is 3). There is only one type of CRIT embedded file, the type generated at problem initialization (i.e. when ICALL is 1). All type two PRED embedded files for the problem are below all CRIT embedded files, and all CRIT embedded files are below all type one PRED embedded files. The first of the PRED and CRIT embedded files for the problem is embedded after the last of the problem summary pages generated by NONMEM for the problem. There are no NONMEM print records dispersed among the records of the PRED or CRIT embedded files for the problem, provided no summarization of iterations from the Estimation Step is printed. If the summarization is printed, then these intermediate pages of output occur between the last page of the last CRIT embedded file (if this file exists, otherwise the last type one PRED embedded file) and the first page of the first type two PRED embedded file (if this file exists). In the event that error message R is issued from the Covariance Step (see section 6.4 of NONMEM Users Guide, Part I), no type two PRED embedded files are printed.

For the purpose of this description then, all type one (two) PRED embedded files for the kth problem may be regarded as a single type one (two) PRED embedded file Pl(k) (P2 (k)). This file could be empty. Similarly, all CRIT embedded files for the kth problem may be regarded as a single CRIT embedded file, C(k). This file too could be empty. From the above it follows that when $k \geq 2$, P2(k) occurs below C(k), C(k) occurs below Pl(k), Pl (k) is embedded after the last of the problem summary pages for problem k, and P2(k) is embedded immediately before the first page of output from the Estimation Step for problem k.

Now similarly, the PRED embedded files for NONMEM initialization (i.e. files generated by PRED when ICALL is 0) should be regarded as a single PRED embedded file, P(0). Also, the CRIT embedded files for NONMEM initialization (i.e. files generated by CRIT when ICALL is 0) should be regarded as a single CRIT embedded file, C(0). Then P2(l) occurs below C(1), C(l) occurs below C(0), C(0) occurs below P1(1), and P1(1) occurs below P(0). Summarization of the iterations from the Estimation Step of the first problem occurs between C(1) and P2(1), if the summarization is output; otherwise there are no NONMEM print records dispersed among the records of these PRED and CRIT embedded files. P(0) is embedded after the last of the problem summary pages of the first problem. If error message R is issued from the Covariance Step, P2(1) is not printed. All of the foregoing is summarized in Table E.5.i.

Table E.5.i Print File

```
PROBLEM Summary
            <---------[P(0)]
            <---------[P1(1)]
            <---------[C(0)]
            <---------[C(1)]
[ITERATION Summary]
…
            <---------[P2(1)] only if message R does not occur
Estimation Step Summary
…
PROBLEM Summary
            <---------[P1(2)]
            <---------[C(2)]
[ITERATION Summary]
            <---------[P2(2)] only if message R does not occur
Estimation Step Summary
…
PROBLEM Summary
…
```

F. Rescaling

In this section the rescaling feature is described. This feature concerns the rescaled canonical parameters (RCP) mentioned in section 5.2 of NONMEM Users Guide, Part I. These are parameters established internally by NONMEM, and the parameter estimate in each summarized iteration from the Estimation Step is given in terms of these parameters. The initial parameter estimate—the one in the summary of the zeroth iteration—is such that all of its components are 0.1. If the number of significant figures specified on the ESTIMATION record is r, then the parameter search terminates when the two estimates resulting from two successive iterations do not differ in the first r significant figures (including leading zeros after the decimal point) in any of the individual parameter components. This criterion applies to estimates in terms of the RCP.

With this background then, suppose a parameter component is initially estimated too large by one or more orders of magnitude. For example, suppose in terms of the RCP its final estimate is 0.00103, whereas its initial estimate is, of course, 0.1. Suppose further that the number of significant figures specified on the ESTIMATION record is 3. Since leading zeros after the decimal point are to be regarded as significant figures, the only figure in the final estimate that follows the leading zeros after the decimal point and that has been accurately determined is that in the thousandth place. Had the initial estimate not been so large, the final estimate could have been more accurate. In this situation the user has the opportunity to continue the parameter search and obtain greater accuracy.

There are two ways this can be done. The first is to continue the search in a subsequent NONMEM run, using the final estimate of the first run as the initial estimate of the second run. The initial estimate in terms of the RCP is now 0.1, and the (accurately determined) significant figures of the final estimate should include no leading zeros after the decimal point. It is recommended that if the user suspects before the first run that the parameter search may need to be continued in a second NONMEM run for this reason, then a Model Specification File be used. (If a Model Specification File is not used, but the initial estimate of the second run (all components considered) is very close to what will be the final estimate of the second run, then the search algorithm may not in fact be able to improve upon this initial estimate due to problems with round-off error. In this case the usual message to this effect is issued.} However, when a Model Specification File is used, the RCP in the second run are the same as those in the first run, and in terms of these the initial estimate of the second run will be no different from the final estimate of the first run, unless the rescaling feature is used. This feature allows the user to specify that a new set of RCP is to be established in the second run, and in terms of these, all components of the initial estimate are 0.1. To so specify, a 1 should be placed in field a of the FIND record. If a blank or zero is placed in this field, rescaling does not occur. (Suppose, though, that a Model Specification File is used, that a 1 is placed in field 4 of the FIND record, but that the Estimation Step is omitted i.e. a 1 is placed in field 1 of the ESTIMATION record. Then a new set of RCP are not in fact established.) The complete format for the FIND record is given in Table F.i.

Table F.i FIND record format ***(See Introduction to NONMEM VI March 2008 for current format.)***

Field No.	**Value**	**Function**
1	between 1 & 9999	identifier for (existing) Model Specification File
2	between 20 & 99	FORTRAN unit number for (existing) Model Specification File
3	0 or blank 1	FORTRAN unit not to be rewound FORTRAN unit to be rewound
4	0 or blank 1	estimate on file not to be rescaled estimate on file to be rescaled

There is a second way the user may continue the parameter search and obtain greater accuracy. He may specify that after the search terminates, a new set of RCP is to be established, in terms of which all components of the final estimate are 0.1, and that the search is to be automatically continued <u>in the same NONMEM run</u>, with the new RCP. This may be accomplished by placing a 1 in field 5 of the ESTIMATION record. The search continuation per se is regarded as a second search, and the usual three lines of summary information from the Estimation Step are given for both first and second searches. Moreover, if iterations are summarized for the first search, then they are summarized for the second search too, but the two summarizations are kept separate, and they are printed on separate pages. The value of r placed in field 3, remains in force during the second search. However, if the upper limit to the number of objective function evaluations that is placed in field 2 is M, while the number of objective function evaluations actually occurring during the first search is n, then the upper limit in force during the second search is M-m. A second search will be implemented only if the first search terminates successfully. In this case if a Model Specification File is to be generated, the final estimate from the second search is output on the file. If the first search terminates unsuccessfully, and if a Model Specification File is to be generated, then the final estimate from the first search is output on the file. The complete format of the ESTIMATION record is given in Table F.ii.

Table F.ii ESTIMATION record format *(See Introduction to NONMEM VI March 2008 for current format.)*

Field no.	**Value**	**Function**
1	0 or blank 1	Estimation Step implemented Estimation Step omitted
If the value is 1, the subsequent fields may be ignored.		
2	nonnegative	Upper limit to number of objective function evaluations
3	between 1 & 8	number of significant figures required in final estimate
4	0 or blank n>0	no monitoring of search routine every nth iteration summarized
5	0 or blank 1	second search not implemented second search implemented
6	0 or blank 1	Model Specification File is not generated Model Specification File is generated
If the value is 0 or blank the following 2 fields may be ignored.		
7	between 0 & 9999	file identifier
8	Between 20 & 99	FORTRAN unit number for file

References

1. Rao, C.R. (1973). Linear Statistical Inference and Its Applications. New York: John Wiley and Sons, pp.625.
2. Anderson, T.W. (1958). An Introduction to Mulitvariate Statistical Analysis. New York: John Wiley and Sons, pp. 374.