The Conditional/Generalized Maximum Likelihood Logit Computer Program: Instructions for Use

Energy Management and Economics

MIT ENERGY LABORATORY REPORT - MIT-EL-78-013

June 1978

PREPARED FOR THE UNITED STATES

DEPARTMENT OF ENERGY

Under Contract No. EX-76-A-01-2295
Task Order 37

Work reported in this document was sponsored by the Department of Energy under contract No. EX-76-A-01-2295. This report was prepared as an account of work sponsored by the United States Government.

The Conditional/Generalized Maximum Likelihood Logit Computer Program: Instructions for Use

Energy Management and Economics

MIT ENERGY LABORATORY REPORT - MIT-EL-78-013

June 1978

PREPARED FOR THE UNITED STATES

DEPARTMENT OF ENERGY

Under Contract No. EX-76-A-01-2295
Task Order 37

TABLE OF CONTENTS

INTRODUCTION

The computer program discussed in Sections A and B was originally developed by Charles Manski. It has since been slightly emended through work by Jerry Hausman, David Wise, Raymond Hartman and Ralph Braid. The latest adaptations reflect efforts aimed at energy demand modeling for the Department of Energy. Section A of this user manual merely duplicates the original Manski user instructions. Section B indicates the most recent version of the program as designed and programmed by Ralph Braid.

A) THE CONDITIONAL/POLYTOMOUS LOGIT PROGRAM: ORIGINAL INSTRUCTIONS FOR USE

1) General Remarks

This FORTRAN program employs the Newton-Raphson algorithm to determine parameter values which maximize the 'conditional logit' likelihood function for a sample of observed choices. The conditional logit model was developed by McFadden (1973) and is a variant of Luce's probabilistic choice model (1959). The expression for the choice probability is as follows:

$$\Pr(i \overset{C}{\varepsilon} C) = e^{Z_i\theta} / \sum_{j\varepsilon C} e^{Z_j\theta}$$ where Z_j is a K-vector of attributes for alternative j, C is an N element choice set and $\overset{C}{\varepsilon}$ designates 'is chosen from'. The K estimated parameters θ are interpretable as the coefficients of a random utility function $u = Z\theta + \varepsilon$, ε a random variable.

The program also provides maximum likelihood estimates for the classical binary logit model and its multi-population extension by Nerlove and Press (1973), among others. In their 'polytomous logit' model, the probability that an observation falls into the i^{th} of N populations is

$$\Pr(i/Y) = e^{Y\phi_i} / \sum_{j=1}^{N} e^{Y\phi_j}$$ where Y is an H component vector of attributes for the observation and ϕ_j is the mean attribute vector for members of the j^{th} population. Given a sample of events, the problem is to estimate ϕ_j, j = 1, ..., N-1, the ϕ_N vector being set to zero

to accomplish a necessary normalization. (Actually, Nerlove and Press use a different normalization. The structure of this program requires the $\phi_N = 0$ one.) In order to perform polytomous logit estimation with this program, the user must make the assignment problem 'look like' a choice problem. This is accomplished as follows:

Define

$$\theta = \begin{bmatrix} \phi_1 \\ \cdot \\ \cdot \\ \cdot \\ \phi_{N-1} \end{bmatrix}$$, a K X 1 vector where K = H *' (N-1)

Define $Z_1 = [Y, \underline{0}, \ldots \underline{0}]$, a series of 1 X K vectors. Here $\underline{0}$ is a 1 X H vector of zeroes.

$$Z_2 = [\underline{0}, Y, \ldots, \underline{0}]$$
$$\vdots$$
$$Z_{N-1} = [\underline{0}, \ldots, Y]$$
$$Z_N = [\underline{0}, \ldots, \underline{0}].$$

We can then write

$$Pr(i/Y) = e^{Y\phi_i} / \sum_{j=1}^{N} e^{Y\phi_j} = e^{Z_i\theta} / \sum_{j \in C} e^{Z_j\theta} = Pr(i \underset{\varepsilon}{} C),$$

the N populations becoming a pseudo N element choice set. After the conversion, the conditional and polytomous logit models are formally equivalent. Only their interpretations differ. The user should be advised that since this program was originally designed for the conditional logit application, the data format required for polytomous logit use is not an efficient one. For conditional logit and classical binary logit, however, it is efficient.

Given the convexity of the conditional logit function, the Newton-Raphson algorithm converges fairly quickly to the maximum whenever one exists. In this program, near extreme collinearity of the data causes a warning to be printed but processing continues. In the case of extreme collinearity, processing terminates. See Jacoby, Kowalik, and Pizzo (1972), among many other books, for details of the Newton-Raphson method.

The program is not internally bounded as to number of parameters to be estimated, number of alternatives in any choice set (choice sets may also vary in size) or in number of observations processed. During processing, data may be stored in core or read in from a secondary device one observation at a time. Being non-conversational, the program can be operated in batch mode or from a terminal.

Time requirements for processing increase fairly linearly with the number of observations, the average size of choice sets and the number of iterations performed and as the square of the number of parameters. As a rule of thumb, each iteration on a purely binary logit problem requires twice the time needed for a linear regression having the same number of observations and parameters. Unless data are stored in core, core requirements remain well under 100K.

The program utilizes two IBM FORTRAN Scientific Subroutine Package routines, DSINV and LOC. If the SSP is not available, these routines must be supplied by the user or replaced by analogous ones.

2) Structure of the Program - User Supplied Routines and Inputs

a. Main Program

The main program dimensions vectors, establishes common blocks and calls the subroutines which perform the computations and bookkeeping. The appropriate main program structure is specified below:

```
COMMON/PRM/TOL,SQTOL,K,NN,ITEND,EPS,IVERGE,IFIRST,ITER,KK,N,ISET
COMMON/KKKK/K
COMMON/BETA/B
COMMON/DBETA/DB
COMMON/NBETA/BNEG
COMMON/NDBETA/DBNEG
COMMON/ERROR/BERR
COMMON/MOMENT/XX
COMMON/EXHWY/XY
COMMON/XE/EXB,XEXB
COMMN/XXE/XXEXB
COMMON/DATCOM/DATA
COMMON/NALT/KCASE
COMMON/VARNM/VNAME
DIMENSION DATA (  ),KCASE(  ), VNAME(  )
DIMENSION B(  ),DB(  ),BNEG(  )DBNEG(  ), BERR(  )
REAL *8 XX(  ),XXEXB(  ),XY(  ),XEXB(  )EXB
CALL BEGIN
1 CALL ITRAT
CALL COMP
CALL CALC(&1)
STOP
END
```

Vector dimensions are as follows. Let K be the number of parameters in the logit model. Let KK = K*(K+1)/2. Then

```
DIMENSION B(K),DB(K),BNEG(K),DBNEG(K),BERR(K),VNAME(2K)
REAL*8  XX(KK),XXEXB(KK),XY(K),XEXB(K),EXB
```

The dimensions for vectors DATA and KCASE are explained in Section 2d (SUBROUTINE READER).

b. Input Parameters

The user inputs, by card if in batch mode and from the terminal if

in conversational model, values for a set of program parameters. These include

TOL - used in test for algorithm convergence. See parameter IVERGE.

SQTOL - used in test for algorithm convergence. See parameter IVERGE.

EPS = a tolerance for loss of significance in inversion of the log-likelihood sedond derivatives matrix. See IBM SSP routine DSINV for details. On loss of significance, a warning is printed but execution continues.

K = number of parameters in the logit model

NN = number of observations in the sample

ITEND = maximum number of iterations to be performed

IVERGE = options on test for algorithm convergence.

If IVERGE = 1, the test $|\Delta B_i| \leq$ TOL, i=1, ..., K is performed.

If IVERGE = 2, the test $1/K * \sum_{i=1}^{K} (\Delta B_i)^2 \leq$ SQTOL is performed.

If IVERGE = 3, both tests are performed and convergence requires satisfying both.

IFIRST = If IFIRST = 1, the first iteration uses zero as an initial value for the logit parameter vector. This option allows fast computation of the first iteration. The option IFIRST = 0 signals that the user will supply an initial parameter vector through SUBROUTINE COEF. This provides the capability for interrupting computation after ITEND iterations and then resuming it later.

ISET = If the observations are stored in core, set ISET = 0. If the observations are stored on a secondary device, set ISET equal to the data set number assigned.

The above parameters should be supplied in the sequence TOL, SQTOL, EPS, K, NN, ITEND, IVERGE, IRFIRST, ISET. The format 3F5.4, 6I5 is required. In conversational mode, the user is prompted when the parameters are to be input.

c. SUBROUTINE COEF

This subroutine must be supplied only if the option IFIRST = 0 is selected. If so, COEF should contain the common block COMMON/BETA/B and read the negative of the initial parameter vector into B.

d. SUBROUTINE READER

READER reads the raw sample data and transforms it, if required, into a form compatible with the program's requirements. READER should contain the common blocks

COMMON/PRM/TOL, SQTOL, K, NN, ITEND, EPS, IVERGE, IFIRST, TER, KK, N, ISET

COMMON/DATCOM/DATA

COMMON/NALT/KCASE

If the option ISET = 0 was selected, the user-supplied subroutine READER is called at once and must, at that time, load the full sample information into the vectors DATA and KCASE. If the secondary storage option was selected, READER is called NN times and must load one observation at a time into DATA and KCASE. The dimensions for DATA and KCASE depend on the option selected.

i) Core Storage Option - Here KCASE is a vector of length NN such that KCASE(t) is the number of alternatives minus one in the t^{th} observation. DATA is a vector whose length can be anything equal to or greater than

$K * \sum_{t=1}^{NN} KCASE(t)$. Let ${}_tZ_{ik}$ be the k^{th} attribute of the i^{th} inferior alternative for observation t. Let c designate the chosen alternative. Then the structure of DATA is as follows:

DATA(J) = $t^{Z}ck - t^{Z}ik$, where

$$J = K * \left[\sum_{h=1}^{t} KCASE(h)\right] - KCASE(t) + K * (i - 1) + k$$

and where t = 1, ..., NN, i = 1, ..., KCASE(t) and k = 1, ..., K.

ii) Secondary Storage Option - Here KCASE is a vector of length (1) such that KCASE(1) is the number of alternatives minus one in the observation currently being processed. Let M equal the maximum value of KCASE over the sample. Then DATA should be dimensioned to have length M * K or greater. The structure of DATA is as follows:

DATA(J) = $t^{Z}ck - t^{Z}ik$, where

J = K * (i - 1) + k and where i = 1, ..., KCASE(1), and k = 1, ..., K.

3) Structure of the Program - Provided Routines and Outputs

The provided program consists of five subroutines. Their names, functions and outputs are as follows:

a. BEGIN - requests input of program parameters, calls subroutine COEF if an initial logit parameter vector is supplied, calls READER if data are to be stored in core, initializes an interation counter and prints a heading.

b. ITRAT - prepares for commencement of an iteration by initializing various variables used later and by updating others. If data are stored on a secondary device, ITRAT rewinds the data set.

c. COMP - processes the sample observations, accumulating the log-likelihood's first derivative vector and second derivative matrix. If data are stored on a secondary device, COMP calls READER when each observation is required. On the first iteration, COMP outputs a count of the number of observations (NN) and choices ($\sum_{t=1}^{NN}$ KCASE(t)) processed. In all iterations, it prints the value of the log-likelihood using the parameter estimate obtained from the previous iteration.

d. CALC - calls the IBM SSP routines DSINV, which inverts the matrix of second derivatives and LOC, which performs

bookkeeping. A check for matrix invertability is made. CALC calculates changes in parameter estimates from the preceding iteration and produces new estimates. It then calls subroutine CONV. If convergence has been obtained or if the maximum number of iterations has been performed, CALC prints the final logit parameter estimates, the change in the estimates from the preceeding iteration and the asymptotic standard errors of the estimates. Control then returns to the main program where execution terminates. If processing is to continue, control is transferred to ITRAT via the main program.

e. CONV - tests for algorithm convergence according to either or both of the two convergence options.

B) EXTENSIONS AND EMENDATIONS TO THE MAXIMUM LIKELIHOOD LOGIT PROGRAM

A few minor changes have been made to the Manski program discussed in Section A to make it slightly more general and easier to use for our purposes. In order to motivate the program changes, we begin with a discussion of the logit log likelihood function, followed by a description of the program changes. This discussion is meant to supplement the Manski writeup.

Logit is a technique designed to model choices made by individuals or other micro units. Suppose the logit program is applied to data at the individual level. Let X_i be the vector of explanatory variables relevant to the choice of the i^{th} individual, which includes the attributes of the various choices and (possibly) individual characteristics. Let k_i denote the choice made by the i^{th} individual. The probability that individual i will make choice k_i is given by

$$P^i_{k_i} = P_{k_i}(X_i, \beta)$$

where β is a vector of coefficients to be estimated. The functions P_j take special forms under the logit assumption (the exact form depending on what kind of logit is assumed), but for the purposes of the present discussion there is no need to write them out explicitly. The likelihood function for a sample of NN observations is given by

$$L(\beta) = \prod_{i=1}^{NN} P_{k_i}(X_i, \beta)$$

and the corresponding log likelihood function is given by

$$\mathcal{L}(\beta) = \sum_{i=1}^{NN} \log P_{k_i}(X_i,\beta) \qquad (1)$$

This is the log likelihood function maximized by Manski's program.

Suppose the logit program is applied to aggregate data. Let there be M groups of individuals (each characterized by a value of the vector X) and let N_{ij} be the number of individuals in group i making the j^{th} choice. Assuming there are J possible choices, the log likelihood function will be

$$\mathcal{L}(\beta) = \sum_{i=1}^{M} \sum_{j=1}^{J} N_{ij} \log P_j(X_i,\beta) \qquad (2)$$

If this were written in the form of equation (1) the necessary value of NN would be

$$NN = \sum_{i=1}^{M} \sum_{j=1}^{J} N_{ij}$$

It follows that if the N_{ij} are on the average quite large, equation (2) is a much more compact way to write the log likelihood function for aggregate data than equation (1).

In order to more easily handle aggregate data, a vector OBS which represents the repetition number of each observation was added to the program. If the program is applied to individual data, then each element of OBS should be set equal to one, and the Manski program will proceed exactly as originally written. If the program is applied to aggregate data, the elements of OBS should be set equal to the appropriate values

of N_{ij} as described in the last paragraph. The program then calculates the log likelihood function and its derivatives in the form of equation (2) rather than equation (1). The main advantage of this change is that it makes the DATA vector somewhat easier to construct and far less wasteful of computer space when the program is applied to aggregate data.

DIMENSION STATEMENTS

The dimension cards used in the various subroutines of this version of the program are listed below. Following the conventions in Manski's program description (Section A), let

K = number of logit parameters

$KK = \frac{K(K + 1)}{2}$

NN = number of observations (individual data)

number of groups * number of choices (aggregate data)

L = number of choices available minus one.

Assuming the core storage option is selected, the appropriate dimension cards are

MAIN PROGRAM

A. DIMENSION DATA (K * L * NN), KCASE(NN), OBS(NN)

B. DIMENSION B(K), DB(K), BNEG(K), DBNEG(K), BERR(K), VNAME(2K)

C. REAL * 8 XX(KK),XXEXB(KK), XY(K), XEXB(K), EXB

SUBROUTINE BEGIN

Cards A, B, and C of MAIN PROGRAM and

REAL * 8 REXB,RREXB, R, XB.

SUBROUTINE READER

Card A of MAIN PROGRAM and one or two additional cards dimensioning

variables necessary for reading the data and constructing the DATA vector.

SUBROUTINE COEF

DIMENSION B(K)

SUBROUTINE ITRAT

Card C of MAIN PROGRAM

SUBROUTINE COMP

Cards A, B, and C of MAIN PROGRAM and

REAL *8 REXB, RREXB, R, XB

REAL * 8 RDATA, RXEXB, LHOOD

SUBROUTINE CALC

Card B of MAIN PROGRAM and

REAL * 8 XX(KK), XY(K), ZY

SUBROUTINE CONV

DIMENSION B(K), DB(K)

The SSP subroutines DSINV and LOC, which are mentioned in Section A, are included in the card deck.

The interpretations of the variables dimensioned in the main program are listed below.

DATA — a vector storing all of the explanatory variables in a manner described in Manski's instructions (section A).

KCASE — number of choices available minus one. If each household chooses among oil, gas, and electric heating, KCASE will be a vector of 2's. KCASE is not necessarily constant across observations in all uses of Manski's program. If

it is not constant, then the DATA vector cannot be dimensioned simply as K * L * NN, since L is not well defined.

OBS — a vector storing the repetition number of each observation, as discussed earlier in this section. Manski did not use this variable in the original version of his program. It would be a vector of all 1's in many applications of the program.

BNEG — logit parameter estimates

B — negative of BNEG

DBNEG — change of BNEG between iterations

DB — negative of DBNEG

BERR — standard errors of coefficient estimates

VNAME — names of explanatory variables

XY — vector storing first derivative of log likelihood function

XX — vector storing second derivative matrix of log likelihood function

XXEXB, XEXB, EXB — variables used in construction XY, XX.

C) LISTING OF PROGRAM

A printout of the program is provided in the following pages, including the main program, all of the subroutines, and the first few data cards. As illustrated here, the program is applied to aggregate data on household heating fuels. The parameters used in forming the dimension statements are

K = 2 — two explanatory variables, fuel price and capital cost

$KK = \frac{K(K + 1)}{2} = 3$

NN = 612 = 204 * 3 — 204 states (rural and urban for two census years) and three choices of heating fuel: gas, oil, and electricity

L = 2 = number of choices minus one.

In applying the program to another problem, the whole READER subroutine must be rewritten. Otherwise, only dimension statements and the data section have to be changed.

Even though it is specific to this problem, the READER subroutine is included for illustrative purposes. The numbers of households in each state using the three different heating fuels are read into the matrix HT(204,3). Fuel prices and capital costs for each state are read into the matrix ZR(204,6), and divided through by the price index DEFL(204). The matrices Z and V, and the vectors K1 and K2, are used in intermediate stages of construction of the data vector.

```
C       CONDITIONAL/POLYTOMOUS LOGIT PROGRAM...CHARLES MANSKI
        COMMON/PRM/TOL,SQTOL,NN,IIEND,EPS,IVERGE,IFIRST,ITER,KK,N,ISET
        COMMON/KKKK/K
        COMMON/BETA/B
        COMMON/DBETA/DB
        COMMON/NBETA/BNEG
        COMMON/NDBETA/DBNEG
        COMMON/ERROR/BERR
        COMMON/MOMENT/XX
        COMMON/EXWHY/XY
        COMMON/XE/EXB,XEXB
        COMMON/XXE/XXEXB
        COMMON/DATCOM/DATA
        COMMON/NALT/KCASE
        COMMON/OBSNUM/OBS
        COMMON /VARNM/ VNAME
        DIMENSION DATA(2448),KCASE(612),OBS(612)
        DIMENSION B(2),DB(2),BNEG(2),DBNEG(2),BERR(2),VNAME(4)
        REAL*8  XX(3),XXEXB(3),XY(2),XEXB(2),EXB
        CALL BEGIN
      1 CALL ITRAT
        CALL COMP
        CALL CALC(&1)
        STOP
        END
```

```
      SUBROUTINE BEGIN
      COMMON/PRM/TOL,SQTOL,NN,ITEND,EPS,IVERGE,IFIRST,ITER,KK,N,ISET
      COMMON/KKKK/K
      COMMON/BETA/B
      COMMON/DATCOM/DATA
      COMMON/NALT/KCASE
      COMMON/OBSNUM/OBS
      COMMON /VARNM/ VNAME
       DIMENSION DATA(2448),KCASE(612),OBS(612)
      DIMENSION B(2),DB(2),BNEG(2),DBNEG(2),BERR(2),VNAME(4)
      REAL*8  XX(3),XXEXB(3),XY(2),XEXB(2),EXB
      REAL*8  REXB,RREXB,R,XB
      PRINT 600
600   FORMAT(1X,'INPUT TOL,SQTOL,EPS,K,NN,ITEND,',','IVERGE,IFIRST,ISET.
     1USE FORMAT(3F5.4,6I5).')
      READ(5,100)TOL,SQTOL,EPS,K,NN,ITEND,IVERGE,IFIRST,ISET
100   FORMAT(3F5.4,6I5)
      WRITE(6,71)TOL,SQTOL,EPS,K,NN,ITEND,IVERGE,IFIRST,ISET
   71 FORMAT(1X,3F5.4,6I5)
C** ** READ VARIABLE NAMES
      DO 10 I=1,K
      K1=(2*I)-1
      K2=2*I
      READ(5,200)(VNAME(J),J=K1,K2)
      WRITE(6,72)(VNAME(J),J=K1,K2)
   72 FORMAT(1X,2A4)
   10 CONTINUE
  200 FORMAT(2A4)
      WRITE(6,73)
   73 FORMAT(' THINGS ARE OKAY HEADING INTO READER')
      IF(IFIRST.EQ.1) GO TO 1
      CALL COEF
1     IF(ISET.EQ.0) CALL READER
      KK=K*(K+1)/2
      WRITE(6,110)NN,ISET,K,IVERGE,ITEND,EPS,IFIRST
      ITER=0
      RETURN
110   FORMAT(///1X,'CONDITIONAL LOGIT ANALYSIS BY',1X,
     1'NEWTON-RAPHSON METHOD'/1X,I5,' DATA RECORDS WILL BE READ FROM',
     11X,'X DATASET',I3,'.'/
     21X,'THE MODEL CONTAINS ',I3,' EXPLANATORY VARIABLES.'/
     31X,'CONVERGENCE OPTION ',I3,' HAS BEEN CHOSEN AND A MAXIMUM',1X,
     4'OF ',I3,' ITERATIONS WILL BE PERFORMED.'/1X,'EPS TOLERANCE',1X,
     5'HAS BEEN SPECIFIED AS ',F7.4,' AND THE INITIALIZATION OF B',1X,
     6'IS HANDLED BY OPTION ',I3,'.')
       END
```

```
      SUBROUTINE READER
      COMMON/PRM/TOL,SQTOL,NN,ITEND,EPS,IVERGE,IFIRST,ITER,KK,N,ISET
      COMMON/KKKK/K
      COMMON/DATCOM/DATA
      COMMON/NALT/KCASE
      COMMON/OBSNUM/OBS
       DIMENSION DATA(2448),KCASE(612),OBS(612)
      DIMENSION ZR(204,6),Z(204,3,2),V(612,2,2)
      DIMENSION HH(204),HT(204,3),DEFL(204),K1(3),K2(3)
      DATA K1,K2/2,3,1,3,1,2/
      READ(5,501) HH,HT,DEFL
  501 FORMAT(10F8.2/10F8.2/10F8.2/10F8.2/10F8.2/F8.2)
C READ IN EXPLANATORY VARIABLES AND CHANGE NOMINAL PRICES TO REAL PRICES
      READ(5,501) ZR
      DO 405 I=1,204
      DO 405 J=1,6
  405 ZR(I,J)=ZR(I,J)/DEFL(I)
      WRITE(6,612)
      WRITE(6,602) HH,HT,DEFL,ZR
C FORM THE Z MATRIX
      DO 411 I=1,204
      DO 411 J=1,3
      DO 412 N=1,2
      M=3*(N-1)+J
  412 Z(I,J,N)=ZR(I,M)
  411 CONTINUE
C DIFFERENCE THE RIGHT HAND SIDE VARIABLES APPROPRIATELY
      M=0
      DO 407 I=1,204
      DO 407 J=1,3
      M=M+1
      OBS(M)=HT(I,J)
      DO 407 N=1,K
      V(M,1,N)=Z(I,J,N)-Z(I,K1(J),N)
      V(M,2,N)=Z(I,J,N)-Z(I,K2(J),N)
  407 CONTINUE
C CONSTRUCT THE DATA VECTOR
      ND=0
      DO 408 NB=1,NN
      DO 408 MB=1,2
      DO 408 LB=1,K
      ND=ND+1
      DATA(ND)=V(NB,MB,LB)
  408 CONTINUE
C CONSTRUCT THE KCASE VECTOR
      DO 409 L=1,NN
      KCASE(L)=2
  409 CONTINUE
C OUTPUT SECTION
      WRITE(6,606)
      WRITE(6,607) DATA
  602 FORMAT(1X,17F7.2)
  606 FORMAT(1H1,'THE DATA VECTOR')
  607 FORMAT(1X,16F7.2)
  612 FORMAT(//1X,'INPUT DATA')
      RETURN
      END
```

```
      SUBROUTINE COEF
      COMMON/KKKK/K
      COMMON/BETA/B
      DIMENSION B(2)
      READ(5,1) (B(J),J=1,K)
    1 FORMAT(5F16.0)
      DO 5 I=1,K
    5 B(I)=-1*B(I)
      RETURN
      END
```

```
      SUBROUTINE ITRAT
      COMMON/PRM/TOL,SQTOL,NN,ITEND,EPS,IVERGE,IFIRST,ITER,KK,N,ISET
      COMMON/KKKK/K
      COMMON/MOMENT/XX
      COMMON/EXWHY/XY
      COMMON/XE/EXB,XEXB
      COMMON/XXE/XXEXB
      REAL*8  XX(3),XXEXB(3),XY(2),XEXB(2),EXB
      EXB=1.D0
      DO 10 I=1,K
      XY(I)=0.D0
   10 XEXB(I)=0.D0
      DO 15 I=1,KK
      XX(I)=0.D0
   15 XXEXB(I)=0.D0
      ITER=ITER+1
      IF(ITER.NE.1.OR.IFIRST.NE.1) IFIRST=0
      IF(ISET.NE.0) REWIND ISET
      RETURN
      END
```

```
      SUBROUTINE COMP
      COMMON /PRM/TOL,SQTOL,NN,ITEND,EPS,IVERGE,IFIRST,ITER,KK,N,ISET
      COMMON /KK<K/ K
      COMMON/BETA/B
      COMMON/MOMENT/XX
      COMMON/EXWHY/XY
      COMMON/XE/EXB,XEXB
      COMMON/XXE/XXEXB
      COMMON/DATCOM/DATA
      COMMON/NALT/KCASE
      COMMON/OBSNUM/OBS
       DIMENSION DATA(2448),KCASE(612),OBS(612)
      DIMENSION B(2),DB(2),BNEG(2),DBNEG(2),BERR(2),VNAME(4)
      REAL*8  XX(3),XXEXB(3),XY(2),XEXB(2),EXB
      REAL*8 REXB,RREXB,R,XB
      REAL*8 RDATA,RXEXB,LHOOD
      ICASE=0
      LCASE=0
      LITER=ITER-1
      LHOOD=0.D0
      NNK=0
      DO 2 II=1,NN
      L=0
      IF(ISET.EQ.0) GO TO 3
      CALL READER
      N=KCASE(1)*K
      GO TO 4
3     N=KCASE(II)*K
4     IF(IFIRST.EQ.1) GO TO 5
      DO 22 JJ=1,N,K
      J1=JJ-1+NNK
      XB=0.D0
      DO 10 I=1,K
      J1I=J1+I
10    XB=XB+DATA(J1I)*B(I)
      IF(XB.LT.-170.D0) GO TO 22
      IF(XB.GT.170.D0) GO TO 13
      GO TO 11
13    JJJ=JJ+NNK
      GO TO 50
11    R=DEXP(XB)
      EXB=EXB+R
      DO 25 I=1,K
      J1I=J1+I
      RDATA=DATA(J1I)*R
      XEXB(I)=XEXB(I)+RDATA
      DO 25 J=1,I
      L=L+1
      J1J=J1+J
25    XXEXB(L)=XXEXB(L)+RDATA*DATA(J1J)
22    L=0
      GO TO 32
5     LCASE=LCASE+1
      DO 42 JJ=1,N,K
      ICASE=ICASE+1
      J1=JJ-1+NNK
      EXB=EXB+1
      DO 35 I=1,K
      J1I=J1+I
      XEXB(I)=XEXB(I)+DATA(J1I)
      DO 35 J=1,I
```

```
      L=L+1
      J1J=J1+J
   35 XXEXB(L)=XXEXB(L)+DATA(J1I)*DATA(J1J)
   42 L=0
   32 REXB=1.D0/EXB
      DO 45 I=1,K
      RXEXB=XEXB(I)*REXB
      XY(I)=XY(I)+RXEXB*OBS(II)
      DO 45 J=1,I
      L=L+1
   45 XX(L)=XX(L)+(XXEXB(L)-RXEXB*XEXB(J))*REXB*OBS(II)
      LHOOD=LHOOD+DLOG(REXB)*OBS(II)
      GO TO 24
   50 DO 51 I=1,K
      JJJ1I=JJJ-1+I
   51 XY(I)=XY(I)+DATA(JJJ1I)
   24 EXB=1.D0
      DO 34 I=1,K
   34 XEXB(I)=0.D0
      DO 44 I=1,KK
   44 XXEXB(I)=0.D0
      IF(ISET.EQ.0) NNK=NNK+N
    2 CONTINUE
      IF(ITER.EQ.1)WRITE(6,140)LCASE,ICASE
      WRITE(6,141) LITER,LHOOD
      RETURN
  140 FORMAT(1X,I6,' CASES CONSISTING OF ',I8,' CHOICES WERE READ')
  141 FORMAT(1X,'THE LIKELIHOOD USING BETA FROM ITERATION ',I5,' IS ',
     XD11.5)
      END
```

```
      SUBROUTINE CALC(*)
      COMMON/PRM/TOL,SQTOL,NN,ITEND,EPS,IVERGE,IFIRST,ITER,KK,N,ISET
      COMMON/KKKK/K
      COMMON/BETA/B
      COMMON/DBETA/DB
      COMMON/NBETA/BNEG
      COMMON/NDBETA/DBNEG
      COMMON/ERROR/BERR
      COMMON/MOMENT/XX
      COMMON/EXWHY/XY
      COMMON /VARNM/ VNAME
      DIMENSION B(2),DB(2),BNEG(2),DBNEG(2),BERR(2),VNAME(4)
      REAL*8 XX(3),XY(2),ZY
      ICIN=0
      IF(IFIRST.NE.1) GO TO 70
      DO 50 I=1,K
   50 B(I)=0.
   70 CALL DSINV(XX,K,EPS,IER)
      IF(IER.NE.0) GO TO 98
   12 CONTINUE
      DO 72 I=1,K
      DB(I)=0.
      DO 72 J=1,K
      CALL LOC(I,J,IR,K,K,1)
      DB(I)=DB(I) -XX(IR)*XY(J)
   72 CONTINUE
      DO 75 I=1,K
   75 B(I)=B(I)+DB(I)
      DO 76 I=1,K
      BNEG(I)=-B(I)
   76 DBNEG(I)=-DB(I)
      CALL CONV(&99)
      GO TO 97
   20 RETURN 1
   99 WRITE(6,110)
      WRITE(6,101) IVERGE
      ICIN=1
      GO TO 15
   98 IF(IER.EQ.-1) GO TO 96
      WRITE(6,102) IER
      GO TO 12
   96 WRITE(6,103)
      RETURN
   97 WRITE(6,110)
  110 FORMAT('1')
      IF(ITER.GE.ITEND) WRITE(6,104)
   15 L=0
      LL=0
      DO 500 I=1,K
      L=L+1
      LL=LL+L
      ZY=XX(LL)
  500 BERR(I)=DSQRT(ZY)
      IF(ITER.GE.ITEND.OR.ICIN.NE.0) WRITE(6,107) ITER
      IF(ITER.LT.ITEND.AND.ICIN.EQ.0) WRITE(6,108) ITER
      WRITE(6,105)
      DO 16 I=1,K
      J1=(2*I)-1
      J2=2*I
   16 WRITE(6,106)I,(VNAME(J),J=J1,J2),BNEG(I),DBNEG(I),BERR(I)
      IF(ITER.LT.ITEND.AND.ICIN.EQ.0) GO TO 20
```

```
      RETURN
101   FORMAT(/////1X,'CONVERGENCE ACCORDING TO CONVERGENCE OPTION ',
     XI3,' HAS BEEN COMPLETED')
102   FORMAT(/////1X,'WARNING. LOSS OF SIGNIFICANCE. IER = ',I3,'. EXECU
     XTION CONTINUES.')
103   FORMAT(/////1X,'EXECUTION TERMINATED. MOMENT MATRIX CANNOT BE INVE
     XRTED OR K HAS BEEN MISSPECIFIED')
104   FORMAT(/////1X,'EXECUTION TERMINATED. MAXIMUM NUMBER OF ITERATIONS
     X HAS BEEN PERFORMED AND THE DESIRED LEVEL OF CONVERGENCE',1X,
     X'HAS NOT BEEN ACHIEVED.')
107   FORMAT(1X,'EXECUTION TERMINATED AFTER',I4,' ITERATIONS.')
108   FORMAT(1X,'ESTIMATED COEFFICIENTS AFTER',I4,' ITERATIONS')
105   FORMAT(//,1X,'VARIABLE',4X,'VARIABLE',11X,'BETA',12X,
     X   'CHANGE IN BETA',6X,
     X   'STANDARD ERROR',/,1X,'NUMBER',6X,'NAME',/,1X,'********',4X,
     X   '*********',9X,'********',10X,'**************',6X,
     X   '**************',/)
106   FORMAT(2X,I3,8X,2A4,8X,E11.5,10X,E11.5,9X,E11.5)
      END
```

```
      SUBROUTINE CONV(*)
      COMMON/PRM/TOL,SQTOL,NN,ITEND,EPS,IVERGE,IFIRST,ITER,KK,N,ISET
      COMMON/KKKK/K
      COMMON/BETA/B
      COMMON/DBETA/DB
      DIMENSION B(2),DB(2)
      IF(IVERGE.EQ.2) GO TO 20
      LP=0
      DO 10 I=1,K
      A=DB(I)/B(I)
10    IF(ABS(A).GT.TOL) LP=1
      IF(LP.EQ.1) GO TO 40
      IF(IVERGE.EQ.3) GO TO 20
      RETURN 1
20    A=0.
      DO 30 I=1,K
30    A=A + DB(I)*DB(I)/B(I)/B(I)
      A=A/K
      A=SQRT(A)
      IF(A.GT.SQTOL) GO TO 40
      RETURN 1
40    RETURN
      END
```

```
      SUBROUTINE LOC(I,J,IR,N,M,MS)
      IX=I
      JX=J
      IF(MS-1) 10,20,30
   10 IRX=N*(JX-1)+IX
      GO TO 36
   20 IF(IX-JX)  22,24,24
   22 IRX=IX+(JX*JX-JX)/2
      GO TO 36 .
   24 IRX=JX+(IX*IX-IX)/2
      GO TO 36
   30 IRX=0
      IF(IX-JX)  36,32,36
   32 IRX=IX
   36 IR=IRX
      RETURN
      END
      SUBROUTINE DSINV(A,N,EPS,IER)
      DIMENSION A(1)
      DOUBLE PRECISION A,DIN,WORK
      CALL DMFSD(A,N,EPS,IER)
      IF(IER) 9,1,1
    1 IPIV=N*(N+1)/2
      IND=IPIV
      DO 6  I=1,N
      DIN=1.D0/A(IPIV)
      A(IPIV)=DIN
      MIN=N
      KEND=I-1
      LANF=N-KEND
      IF(KEND) 5,5,2
    2 J=IND
      DO 4  K=1,KEND
      WORK=0.D0
      MIN=MIN-1
      LHOR=IPIV
      LVER=J
      DO 3 L=LANF,MIN
      LVER=LVER+1
      LHOR=LHOR+L
    3 WORK=WORK+A(LVER)*A(LHOR)
      A(J)=-WORK*DIN
    4 J=J-MIN
    5 IPIV=IPIV-MIN
    6 IND=IND-1
      DO 8 I=1,N
      IPIV=IPIV+I
      J=IPIV
      DO 8 K=I,N
      WORK=0.D0
      LHOR=J
      DO 7  L=K,N
      LVER=LHOR+K-I
      WORK=WORK+A(LHOR)*A(LVER)
    7 LHOR=LHOR+L
      A(J)=WORK
    8 J=J+K
    9 RETURN
      END
      SUBROUTINE DMFSD(A,N,EPS,IER)
      DIMENSION A(1)
```

```
      DOUBLE PRECISION DPIV,DSUM,A
      IF(N-1) 12,1,1
    1 IER=0
      KPIV=0
      DO 11 K=1,N
      KPIV=KPIV+K
      IND=KPIV
      LEND=K-1
      TOL=ABS(EPS*SNGL(A(KPIV)))
      DO 11  I=K,N
      DSUM=0.D0
      IF(LEND)  2,4,2
    2 DO 3  L=1,LEND
      LANF=KPIV-L
      LIND=IND-L
    3 DSUM=DSUM+A(LANF)*A(LIND)
    4 DSUM=A(IND)-DSUM
      IF(I-K)  10,5,10
    5 IF(SNGL(DSUM)-TOL)  6,6,9
    6 IF(DSUM)  12,12,7
    7 IF(IER)  8,8,9
    8 IER=K-1
    9 DPIV=DSQRT(DSUM)
      A(KPIV)=DPIV
      DPIV=1.D0/DPIV
      GO TO 11
   10 A(IND)=DSUM*DPIV
   11 IND=IND+I
      RETURN
   12 IER=-1
      RETURN
      END
//G.SYSIN DD *
.001  .001  .001      2  612   15     3     1     0
        D
      CAP
```

References

1. Hausman, J. and Wise, D., "A Conditional Probit Model for Qualitative Choice: Discrete Heterogeneous Preferences," forthcoming in Econometrica.

2. Jacoby, Kowalik, and Pizzo, Iterative Methods for Non-Linear Optimization Problems, Prentice Hall, 1972.

3. McFadden, "Conditional Logit Analysis of Qualitative Choice Behavior, in Zerembka, ed., Frontiers in Econometrics, Academic Press, 1973.

4. Manski, "The Conditional/Polytomous Logit Program: Instructions for Use," Mimeo, Carnegie-Mellon University, July 1974.

5. Nerlove and Press, "Univariate and Multivariate Log-Linear and Logistic Models," Rand Report No. R-1306-EDA/NIH, 1973.