HLOGIT.DOC

--HLOGIT.DOC--

HIERARCHICAL CHOICE MODELS: THREE LEVEL NESTED MULTINOMIAL LOGIT

PROGRAM (VERSION 6.2): NOV 29, 1989
Documentation Update: FEB 16, 1990

---

1. INTRODUCTION

1.1 GENERAL DESCRIPTION:

This program analyses the nested multinomial logit model (NMNL) or hierarchical decision tree with up to three levels.

In particular, HLOGIT will

- Define and display any one to three-level tree structure;
- Check the data and calculate sample means and frequencies;
- Generate interactions between variables and alternatives;
- Weight observations for grouped data estimation;
- Allow for choice based sampling (WESML);
- Support unbalanced choice sets;
- Support any parameter equality restrictions;

- Maximize the full information likelihood function;

- Predict market shares and compute prediction success table;
- Calculate the covariance matrix correcting for WESML sampling;
- Calculate elasticities of choice probabilities;
- Test the significance of the tree structure;
- Compute implied correlation matrix of unobserved utility components.

The Program runs as batch-program or interactively. In the latter case, there is a menu-display before each operation and you will be prompted for options at various stages of the program. See Chapter 3 for details.

The user has to provide three input files

"HLOGIT.PRO" (explained in Section 2.1):
Profile with default values. In batch mode, commands are appended.

"INPARMS.PAR" (explained in Section 2.2):
Dimensions, tree structure, parameter labels, and initialvalues.

"INDATA.DAT" or "INDATA.BIN" (explained in Section 2.3):
Data (dependent and explanatory variables by alternative) in
ASCII or binary form.

In interactive mode, the terminal will prompt for actual file names.
A blank name is interpreted as usage of the default names specified in
"HLOGIT.PRO" which will also be used in batch mode.

Results will be printed on the terminal screen. They can also be echoed ("spooled") on a file using the spool option. In addition, the estimated parameters are always written on a file that has the same structure as "INPARMS.PAR" and can therefore be used as input at a later stage.

1.2. ECONOMETRIC SPECIFICATION AND NOTATION

As an example for notation and terminology, consider the following NMNL model symbolized by a decision tree:

```
  Level 0: (stem)                                   |
                                           +-----------------+
  Level 1: (limbs)                         1                 2
                                           |                 |
     Upper-level dissimilarity coeff.     ()           +---()-------+
                                           |            |             |
  Level 2: (branches)                      11           21            22
                                           |            |             |
     Lower-level dissimilarity coeff.    +[]+     +-+-[]-+-+     +-[]-+
                                         | | |     | | | | | |     | | | |
  Level 3: (twigs)                       | | |     | | | | | |     | | | |
                                         | | |     | | | | | |     | | | |
     Elemental Alternatives:             1 2 3     4 5 6 7 8 9     0 A B C
```

This tree has two limbs. The first limb has one upper-level dissimilarity parameter (c,). This dissimilarity parameter is called "trivial" because the limb has only one branch. This branch has one (non-trivial) lower-level dissimilarity parameter (e') and three elemental alternatives.
The second limb has one (non-trivial) upper-level dissimilarity parameter (c,) and two branches. Each of these branches has one (non-trivial) lower-level dissimilarity parameter (e'). The first branch has six, the second branch four elemental alternatives.

We will employ a short-hand notation for trees consisting of numbers and letters symbolizing elemental alternatives, separated by commas and semicolons to denote the structure of branches and limbs. Semicolons separate limbs, commas separate branches from each other.

The above tree will be denoted by the string 123;456789,0ABC because alternatives 1-3 are on a different limb as alternatives 4-C and alternatives 4-9 on a different branch as alternatives 0-C.
The corresponding MNL model would be simply a string of all 13 alternatives: 1234567890ABC.

Alternatives are labeled by numbers and letters as follows:
Labels 1-9 for alternatives 1-9
Label 0 for alternative 10
Labels A-Z for alternatives 11-36

```
  LIKELIHOOD FUNCTION:
  --------------------

The NMNL model is characterized by the likelihood function

   L = log { P(T|B,L) * P(B|L) * P(L) } , where


   T=chosen twig, B=chosen branch, L=chosen limb

   P(T|B,L) = exp { X(L,B,T)*BETA/THETA(L,B) } / exp { INC2(L,B) }

           where INC2(L,B) = log  a"  exp { X(L,B,T’)*BETA/THETA(L,B) }
                                T’i^ B
                                B i^ L

  P(B|L) = exp { INC2(B)*THETA(L,B)/TAU(L) } / exp { INC1(L) }

           where INC1(L) = log  a"  exp { INC2(B’)*THETA(L,B’)/TAU(L) }
                              B’i^ L

  P(L) = exp { INC1(L)*TAU(L) } / exp { INC0 }

           where INC0 = log  a"  exp { INC1(L’)*TAU(L’) }
                             L’

NOTE:  We employ the so-called "discrete choice formulation" in which
=====  the parameters BETA are constant across alternatives.

IMPORTANT: To be identified, the corresponding variables X(L,B,T)
  must exhibit at least some variation across alternatives
  ("constant-BETA/varying-X description").
  Handling of agent-specific characteristics that do not vary across
  alternatives is discussed in Section 1.3 on the interaction feature.

---------------------------------------------------------------------------
```

1.3 THE INTERACTION FEATURE

A user should be aware that discrete choice models can include two different kinds o

- variables that vary across alternatives ("alternative-specific attributes")

- variables that are constant across alternatives and characterize the decision-making agent ("agent-specific characteristics")

HLOGIT provides a convenient way to interact alternatives and both kinds of explanatory variables. All variables can interact with one or more dummy variables

The choice among a finite set of discrete alternatives is governed by the DIFFERENCES among the alternatives, not the ABSOLUTE LEVEL of their attributes. Attributes that are constant across alternatives do not differentiate the alternatives, they are thus irrelevant for the choice among them. Coefficients associated with constant attributes cannot be identified.

However, agent-specific characteristics may affect different alternatives in different ways. In other words, there is an interaction effect between this agent-specific characteristic and some alternatives RELATIVE to other alternatives. In order to identify agent-specific characteristics it is therefore necessary to interact agent-specific characteristics with dummy variables that are turned on in some alternatives and turned off in others.

Variables that vary across alternatives ("alternative-specific attributes") are identified without such interaction terms. Sometimes, however, a user may estimate different coefficients for different (groups of) alternatives. Again, this amounts to interacting these variables with dummies that are specific to certain alternatives.

HLOGIT provides a map for each variable (including the constant term) in which these interactions must be specified. For each variable K, the map consists of the number of interaction terms, NXM(K), and an array

LMAP(K,I,M) that defines whether dummy variable M, M=1,..,NXM(K) is turned on in alternative I, I=1,..,NALT, the number of alternatives.

It is helpful to distinguish the following three cases:

```
(CASE 1) NO INTERACTION:
------------------------
NXM(K)=1,  LMAP(K,I,1)=1   for all alternatives I=1,..,NALT

EXAMPLE for NALT=6:     NXM=1      LMAP=1 1 1 1 1 1
```

In this case, one common coefficient for all alternatives is associated with variable K. This coefficient will only be identified, if variable K exhibits variation across alternatives. This is the standard case for alternative-specific attributes that vary across alternatives (like travel time in transportation mode choice)

```
(CASE 2) STANDARD INTERACTION:
------------------------------
NXM(K)=NALT-1     LMAP(K,I,M)=1   if I=M      I=1,..,NALT; M=1,..,NALT-1
                              0   else

EXAMPLE for NALT=6:     NXM=5      LMAP=1 0 0 0 0 0
                                        0 1 0 0 0 0
                                        0 0 1 0 0 0
                                        0 0 0 1 0 0
                                        0 0 0 0 1 0
```

In this case, NALT-1 constants will be assigned to each elemental alternative except one (here: the last one). NALT-1 coefficients will be estimated for variable K, each measuring the effect of this variable on a single elemental alternative relative to the omitted (here: last) alternative. This is the standard case for agent-specific characteristics that exhibit no variation across alternatives (like income in mode choice).

(A BRIEF NOTE ON THE SPECIFICATION OF DISCRETE CHOICE MODELS:

HLOGIT adopts the "constant-BETA/varying-X" formulation of discrete choice models in the above likelihood function. For variables that does not vary across alternatives, the standard interaction with NALT-1 dummies replaces

```
     X(L,B,T)*BETA
```

in the above "constant-BETA/varying-X" likelihood function by the interactions

```
     [X*DUMMY(L,B,T)]*BETA = X*[DUMMY(L,B,T)*BETA].
```

This is evidently equivalent to estimating a "constant-X/varying-BETA" likelihood function with

```
      X*BETA(L,B,T)
```

in the place of X(L,B,T)*BETA.)

(CASE 3) USER-SPECIFIED INTERACTIONS:

Instead of assigning NALT-1 dummies where each dummy corresponds to exactly one alternative, one may want to assign a set of NXM(K)NALT-1 dummies where each dummy may represent one or more alternatives. In this case, NXM(K) coefficients will be estimated for each variable K.

NOTE: This procedure amounts to placing simple linear restrictions on the standard interactions in CASE 2.

NOTE: This procedure is particularly useful for a symmetric tree where the alternatives have a factorial structure, that is, each limb has the same number and kind of branches, and each branch has the same number and kind of alternatives. In this case, (number of limbs-1)+(number of branches per limb-1)+(number of alternatives per branch-1) constants will be assigned, implying an assignment of less than NALT-1 constants, therefore a substantial saving in the number of explanatory variables.

NOTE: In the case of variables that do not vary across alternatives, the user should of course be careful that the dummies do not add up to one.

EXAMPLES for NALT=6, TREE=12,3;45,6:

```
      a) NXM=3    LMAP=1 1 1 0 0 0
                       1 1 0 1 1 0
                       1 0 0 1 0 0
```

In this example, dummy 1 characterizes the first limb, dummy 2 the first branch in each limb, and dummy 3 the first alternative in branches 11 and 21. The first coefficient of each agent-specific characteristic describes the effect of that characteristic on all alternatives in the first limb. The second coefficient of each agent-specific characteristic describes the effect of that characteristic on the alternatives in the first branches in each limb. The third coefficient of each agent-specific characteristis describes the effect of that characteristic on the first alternatives in each limb. All effects are relative to the last alternative.

```
      b) NXM=4    LMAP=1 1 1 0 0 0
                       1 1 0 0 0 0
                       0 0 0 1 1 0
                       1 0 0 0 0 0
```

In this example, dummy 1 characterizes the first limb, dummy 2 the first branch in limb 1, dummy 3 the first branch in limb 2, and dummy 4 alternative 1 only.

```
NOTE:  See also the example in data section 2.3 that shows the correspondence
=====  between raw data (without interactions) and expanded data (interacted).
```

2. USAGE OF HLOGIT

HLOGIT can be called in four ways from the DOS-prompt:

```
  C:>HLOGIT
  C:>HLOGIT MYPROFIL.TXT
  C:>HLOGIT /B
  C:>HLOGIT /B MYPROFIL.TXT
```

The first two calls invoke the interactive version of HLOGIT, the last two calls invoke the batch version. Calls 1 and 3 use the standard profile with name HLOGIT.PRO. Calls 2 and 4 use a user-defined profile with name MYPROFIL.TXT.

The following sections describe the three input data sets "HLOGIT.PRO", "INPARMS.PAR", and "INDATA.DAT/BIN" that must be present for HLOGIT.

2.1 THE PROFILE "HLOGIT.PRO":

The profile contains the default values for parameters and filenames.
If no profile is specified in the command line, HLOGIT uses the default profile with the name HLOGIT.PRO.

In interactive mode, the profile defines the default values of data handling and optimization which may be overridden in the actual prompts.

In batch mode, the profile contains the actual values of data set names and the batch commands.

The profile contains at least 11 lines. The first line is a header which is ignored by the program. In the following 10 lines, 10 default values are attached to 10 keywords:

```
    keyword:  value:       default for:
    scr_pause=5            length of pause (in seconds) between screens
    max_obs  =0            maximum number of observation (0=EOF or buffer size)
    data_mode=0            data mode: 0=ascii, 1=binary, 2=ascii&binary scratch
    max_iter =10           iteration limit for optimizer
    accuracy =0.001        accuracy for optimizer
    optimizer=2            algorithm for optimizer
    param_in =hltest.par   file name for parameter input file
    data_in  =hltest.dat   file name for data input file
    spool_out=hltest.spo   file name for spool output file (blank=none)
    param_out=hltest.out   file name for parameter output file
```

Do not change the order of the lines, and start each input in column 11, as shown above. Comments are allowed past column 21, except for the last four rows of filenames which may be up to 40 characters long.

scr_pause: Length of pause (in seconds) between screens in interactive mode.
0 or smaller: no pause
1-59: pause in seconds
60 or larger: prompt between screens
(In batch mode, scr_pause is ignored.)

max_obs: Number of observations used in estimation.
0 denotes as many observations as the internal workspace will hold.
If the actual data set is smaller than any of the above, the entire data set is used.

data_mode: type of data set, see Section 2.3

max_iter, accuracy, optimizer: iteration limit, accuracy and algorithm of optimaization routine, see Section 3.1

param_in: default name for input parameter file "INPARMS.PAR". Must be present. See Section 2.2

data_in: default name for data input file "INDATA.DAT". Must be present. See Section 2.3

spool_out: default name for spool output file. Optional. If not blank, all screen-output is also echoed to this file. See Section 3.12.

param_out: default name for parameter output file "OUTPARMS.PAR". Must be present. See Section 3.20

In batch mode, the profile then contains (starting in line 12) the batch commands described in Section 3. See HLDEMO.PRO for an example of a profile which includes batch commands, and HLOGIT.PRO for an example of a standard profile without batch commands.

2.2 TREE-SPECIFICATION AND INITIAL-VALUES: THE INPUT FILE "INPARMS.PAR":

This file contains

A) the functional form of the decision-tree,
   i.e., the arrangement of the alternatives in the tree,
B) the number of explanatory variables,
C) the interactions for each explanatory variable,
D) and the initialvalues for the parameters.

Input lines are marked by in the sequal by ">>>>>" in the margin. They are read by list-directed I/O (*-format). Therefore, inputs must be separated by a comma and/or at least one blank space.

A) TREE STRUCTURE   [format 72A1]

```
>>>>>   STRUCT(1,...,72)
```

The tree structure is denoted by the string STRUCT of up to 72

```
        characters.  Numbers 1-9, 0, then letters A-Z in this string symbolize
        the elemental alternatives in that order.
        Semicolons separate limbs, commas separate branches from each other.
        See the example at the top of this file and the test program at the
        bottom of this file to get familiar with this notation.

        NOTE: HLOGIT checks the level of the tree by counting
              commas and semicolons in the TREE string.
              Case 1: If there is at least one semicolon, the full three-level
                      code is used.
              Case 2: If there is at least one comma, but no semicolon, the
                      quicker two-level code is used.
              Case 3: If there is neither a comma nor a semicolon, simple MNL
                      is applied.
              The number of alternatives is computed as the number of non-blank
              characters in STRUCT that are not commas or semicolons.

        NOTE: The order of the alternatives in STRUCT is significant and can
              be different from 1-9,0,A-Z. The data, however, must always be
              in ascending order 1-9,0,A-Z.  Also any other input in this file
              refers to the alternatives in ascending order, not the actual
              order specified in STRUCT.  The only exception are the initial
              values for the dissimilarity parameters which refer to the actual
              order as specified in STRRUCT, see below.


     B) NUMBER OF EXPLANATORY VARIABLES  [*-format]

>>>>>   NX, NXD, NXA, NWT, NCS

        where NX  = number of alternative-specific attributes
                       (these variables vary across alternatives)
              NXD = number of agent-specific characteristics
                       (these variables do not vary across alternatives and
                        must interact with at least one dummy that varies
                        across alternatives).
              NXA = 0 if a constant should not be included.
                  = 1 if a constant should be included,
                (the constant does not vary across alternatives and
                 must interact with at least one dummy that varies
                 across alternatives).
        NWT = 0 if each observation has the same weight,
            = 1 if weights are included,
        NCS = 0 if each observation has the same choice set.
            = 1 if a choice set variable is included,
```

NOTE: NX, NXD refer to the number of variables read from the data file. Do not count interactions, and do not distinguish between those variables that are associated with coefficients that will be estimated freely and those with equality-constrained coefficients.

NOTE: Cf. Subsection 2.3.2, Variable Types, for further discussion.

C) INTERACTIONS [*-format]

```
>>>>>   For each variable K = 1 through NX+NXD+NXA:
>>>>>
>>>>>   NXM(K)     LMAP(K,1,1)  , ... ,LMAP(K,NALT,1)
>>>>>              ...
>>>>>              LMAP(K,1,NXM), ... ,LMAP(K,NALT,NXM)
```

```
        where NXM(K)      = number of dummy variables interacting with
                                    variable K
              LMAP(K,I,M) = 1 if the M-th dummy for variable K is turned on
                                    in alternative I,
                            0 if turned off.
```

NOTE: See Section 1.3 for examples.

NOTE: The order of variables K as well as the order of alternatives I in LMAP has to correspond to the order of the variables and the alternatives in the data. Specifically, the first NXM(1) rows in LMAP relate to the first variable, the second NXM(2) rows relate to the second variable, etc. Similarly, the first column in LMAP relates to alternative labeled 1, the second column relates to alternative labeled 2, etc.

NOTE: The total number of lines in the maps for all variables has to be equal to the number of initial values for all parameters (excl. dissimilarity parameters), NXX. NXX is the sum of NXM(K) for all K=1,..,NX+NXD+NXA, i.e., the number of all interactions (incl. the trivial ones) of all explanantory variables (alternative-specific variables, agent-specific variables and alternative-specific constants).

D) INITIAL VALUES [format (A8,1X,I1,F15.9)]

```
>>>>>   For each parameter J=1,..,NPP:
>>>>>
>>>>>   column 1...8:   column 10:   column 11...24:
>>>>>
>>>>>   NAME( 1)        ISTAT( 1)    PARM( 1)
```

```
where: NAME(J)  = label of the J'th variable/parameter,

       ISTAT(J) = 0 if the corresponding parameter should be estimated
                    freely,
                 -1 if the corresponding parameter is constrained to a
                    fixed number,
                  m if the corresponding parameter is constrained
                    to be equal to another parameter with the same
                    number m ("inter-parameter constraint").  Each m>0
                    creates a set of equality-constrained parameters.

       PARM(J)  = initialvalues of the J'th parameter,

       NPP = total number of all primary parameters = NXX + NTH + NTAU.
       (See below for the definition of "primary" parameters)
```

NOTE: The user must be careful

- to include the correct number of initial values:
    1) sum of all NXM(K) interactions of all NX+NXD+NXA explanatory variables K (=NXX)
    2) plus all nontrivial dissimilarity parameters (=NTH+NTAU)

- to respect the order of the initial values:
    1) Coefficients of explanatory variables (BETA)
        a) Interactions of alternative-specific attributes
        b) Interactions of agent-specific characteristics
        c) Interactions of alternative-specific constants
    2) Upper-level dissimilarity parameters (level 1) (TAU)
    3) Lower-level dissimilarity parameters (level 2) (THETA)

The order of the explanatory variables has to correspond to the order in the data file.

NOTE: The number of initialvalues for explanatory variables (excl. dissimilarity parameters), NXX, has to be equal to the number of lines in the interaction maps.

The order of the dissimilarity parameters is from the left to the right in the tree as specified in STRUCT.

NOTE: For each non-trivial dissimilarity parameters one initial value must be provided.
Trivial upper-level dissimilarity parameters are in limbs that have only one branch, trivial lower-level dissimilarity parameters are in branches that have only one alternative. They do not get initialvalues.

HLOGIT distinguishes three types of parameters that correspond to the three different ISTAT values (0, -1, and m>0):

1) Parameters that are completely free (ISTAT=0). This is the standard status for likelihood maximization. On estimation, they are chosen to maximize the loglikelihood function.
2) Parameters that are fixed at a given number (ISTAT=-1). On estimation, their value is kept fixed.

3) Parameters that are being estimated but that are also constrained to be equal to some other parameter (ISTAT=m>0). Each m>0 defines a set of equal parameters. On estimation, their joint value is chosen to maximize the loglikelihood function.

NOTE: These three parameter types define three parameter vectors:
1) The vector of "primary parameters". It includes all NPP parameters of type 1), 2) and 3) in the "INPARMS.PAR" file. NP=NXX+NTAU+NTH.
2) The vector of "free parameters". It includes only those parameters that are not constrained to a fixed number, i.e., all parameters with ISTAT0, i.e. all parameters of type 1) and 3), counting all parameters subject to inter-parameter equality constraints separatly.
3) The vector of "deep parameters". It includes the NP parameters entering the likelihood maximization. Sets of equality-constrained parameters identified by a common ISTAT>0 count only once. Hence, NP is the number of all parameters with ISTAT=0 plus the number of sets of equality-constrained parameters.

2.3 DATA: THE INPUT FILES "INDATA.DAT" and "INDATA.BIN":

HLOGIT is intended to be addressed by an external data handling program that generates and manipulates the data that will be fed into HLOGIT for estimation and analysis.

Data can be read in from a file with ASCII numbers or an HLOGIT binary file. ASCII-files are portable, binary files are much faster to read and write. HLOGIT binary files should be created by HLOGIT if the same data set will be used several times.

We will first describe some generalities, then the format of an ASCII raw data file and an HLOGIT binary data file.

2.3.1 SIZE OF DATA SET:

HLOGIT does not limit the number of observations. However, if the data exceeds the size of the internal worksheet ("buffer"), data has to be moved between the HLOGIT binary scratch file and memory ("paging") which substantially slows the program down.

The size of the internal worksheet to store data depends on the specific installation of HLOGIT. In the standard version, the buffer is exactly one memory segment of 64 KBytes, corresponding to MAXBUF=16000 numbers (see the beginning of Chapter 3).
The number of observations that fit in this internal worksheet depends on the number of alternatives (NALT), the number of explanatory variables (NX and NXD), and whether a weight and a choice set variable are included or not (NWT and NCS). It can be computed by the formula:

```
   MAXOBS = MAXBUF/NDA - 2
```

where the number of data items per observation is

```
   NDA = (NX+NCS)*NALT+1+NWT+NXD
```

All data is read at the beginning of the program. If the file is ASCII, the user has at this point the option of generating an HLOGIT data file in binary format. The creation of this binary file is required if the data set does not fit into memory.

2.3.2 VARIABLE TYPES:

Six types of variables should be distinguished (Cf. Section 1.3):

(1) The dependent variable. For each observation, this is the index of the chosen alterative. (See below for indexation.)

(2) If NX > 0:
NX alternative-specific explanatory variables ("attributes") which vary by NALT alternatives and (possibly) by observation. For each observation, this is a two-dimensional array of NALT rows and NX columns.

(3) If NXD > 0:
NXD agent-specific variables ("characteristics") which vary only by observation but not by alternative. For each observation, this is a row-vector of length NXD.

(4) If NXA = 1:

A constant term. It will be generated internally, so one must not include it in the data.

(5) If NWT = 1:
The weights. For each observation, this is a number carrying a weighting factor. (See below for examples.)

(6) If NCS = 1:
The choice set variable. For each observation, this is a column vector of zeroes and ones of length NALT, in which a one (zero) indicates that the corresponding alternative is included (excluded) in the observation's choice set.

NOTE: In oder to conserve space in HLOGIT data sets, the above order of variables does NOT correspond to the actual order in HLOGIT data sets. For the actual order, see Subsection 2.3.3 below.

The dependent variable indicates the alternative chosen and carries a value between 1 and NALT, corresponding to the order in which the alternatives are stored.

The weights are applied to each observation's likelihood as well as to summary statistics, predictions and elasticities. They can also be used as replication factors for grouped data analysis since the sum of the weights across observations need not sum up to one. In all summary statistics, HLOGIT divides by the sum of weights, not the number of physical observations.

EXAMPLES:

(1) In simple random sampling with balanced choice sets, the weights in each observation and alternative are 1.0. Therefore, it is unnecessary to include weights in the data, set NWT=0.

(2) In simple choice based sampling, the weights are the population share of the alternative chosen by the observation divided by the sample share of this alternative.

(3) In stratified random sampling, the weights represent the weight of the corresponding stratum.

(4) The user can combine (2) and (3).

2.3.3 ASCII raw data files:

The first line of the ASCII data file must be a valid REAL FORTRAN format description. The format has to be enclosed in parentheses. Including the parentheses, the format must not exceed 72 characters.

The program then expects NALT lines for each observation. Each line carries NCS+NX+NWT+1+NXD entries in the following order using the above user-specified format.

1) 1, if the corresponding alternative is included in the choice set of the observation, 0 else. (Omit this entry, if NCS=0).
2) NX alternative-specific characteristics for the corresponding alternative and observation. (Omit these entries, if NX=0).
3) The dependent variable (=index of the chosen alternative. This entry must always be present)
4) The weight (Omit this entry, if NWT=0)
5) NXD agent-specific attributes. (Omit these entries, if NXD=0).

NOTE: Items 3)-5) are the same for all alternatives of an observation.

NOTE: For each observation, the first line is read completely with the complete format description. In the remaining NALT-1 lines of each observation, only the first NCS+NX items are read with the first NCS+NX items of the above format. The index of the chosen alternative and the NXD agent-specific attributes are ignored, since they are identical to those in the first alternative. (Accordingly, they can also be omitted in the data file, creating a non-rectangular file structure).

NOTE: If NCS+NX=0, each observation must have only one line of data.

NOTE: The alternatives are assumed to be in ascending order independently of the structure of the tree that will be estimated. The dependent variable must be an integer between 1 and NALT corresponding to the alternatives in ascending order. In other words, the index of each alternative is defined by the order in the data file.

NOTE: The number and the order of alternatives must always be the same for each observation. This should be emphasized in the case of unbalanced choice sets. In this case, also the alternative-specific variables in the alternatives excluded from the choice set should be given some value to avoid a read error although this value will never be used.

EXAMPLE: Imagine the following data for a four-alternative choice problem (NALT=4):

1) A choice set variable (assume, observation one has no alternative 3) (NCS=1).
2) Two alternative-specific attributes P ("price of alternative") and Q ("quality of alternative"). P should be attached with two different coefficients, one for alternatives 1 and 2, the other for alternatives 3 and 4. Q should be attached with a common coefficient in all four alternatives:
```
    (NX=2,  NXM=2, LMAP=1 1 0 0
                        0 0 1 1
            NXM=1, LMAP=1 1 1 1)
```
3) The dependent variable (Assume, observation 1 has chosen alternative 2 and observation 2 has chosen alternative 3)
4) Weights are included (assume, observation 1 has a third of the weight of observation 2) (NWT=1)
5) Three agent-specific characteristics Y,S,A ("income, household size, and age") which are interacted by two dummies
```
    (NXD=3, NXM=2, LMAP=1 0 0 0
                        0 1 1 0)
```
6) A constant which is interacted with three dummies
```
    (NXA=1, NXM=3, LMAP=1 0 0 0
                        0 1 0 0
                        0 0 1 0)
```

```
The ASCII-data file should look like:     or, more economical, like:

(F3.1,2F10.3,2F5.1,3F10.3)                 (F1.0,2F9.3,F2.0,F4.1,3F9.3)
1.0  P11  Q11  2.0  0.5  Y1  S1  A1        1 P11 Q11 2 0.5 Y1 S1 A1
```

```
1.0  P12  Q12  2.0  0.5  Y1  S1  A1       1 P12 Q12
0.0  P13  Q13  2.0  0.5  Y1  S1  A1       0 P13 Q13
1.0  P14  Q14  2.0  0.5  Y1  S1  A1       1 P14 Q14
1.0  P21  Q21  3.0  1.5  Y2  S2  A2       1 P21 Q21 3 1.5 Y2 S2 A2
1.0  P22  Q22  3.0  1.5  Y2  S2  A2       1 P22 Q22
1.0  P23  Q23  3.0  1.5  Y2  S2  A2       1 P23 Q23
1.0  P24  Q24  3.0  1.5  Y2  S2  A2       1 P24 Q24
etc.                                      etc.
```

```
NOTE: The interaction feature will expand the explanatory variables
      internally to the following array:

      P11  0    Q11  Y1  0   S1  0   A1  0   1  0  0
      P12  0    Q12  0   Y1  0   S1  0   A1  0  1  0
      0    P13  Q13  0   Y1  0   S1  0   A1  0  0  1
      0    P14  Q14  0   0   0   0   0   0   0  0  0
      P21  0    Q21  Y2  0   S2  0   A2  0   1  0  0
      P22  0    Q22  0   Y2  0   S2  0   A2  0  1  0
      0    P23  Q23  0   Y2  0   S2  0   A2  0  0  1
      0    P24  Q24  0   0   0   0   0   0   0  0  0
      etc.

      (The PRINT DATA option -- see below, Subsection 3.11 -- will display
      the data in this expanded fashion)
```

2.3.4 HLOGIT binary data files

For speedier execution, the ASCII data file can be converted to an HLOGIT binary file. This file is created by specifying the appropriate option in the prompt before reading the data in. Binary data files must be created for data sets larger than the available workspace.

The HLOGIT binary data files consist of NOBS vectors of
   NDA=(NCS+NX)*NALT+1+NWT+NXD
items. All items are stored as REAL*4 numbers.

For each observation, the data is stacked as follows:
1) NALT values of the choice set variable (only if NCS=1)

```
2) NALT values of the first alternative-specific explanatory variable,
    ...
   NALT values of the NX'th alternative-specific explanatory variable,
3) 1 value of the weight (only if NWT=1)
4) 1 value of the dependent variable,
5) NXD values of the agent-specific variables.

IMPORTANT NOTE:
      The binary data is written in the order as specified in the
      currently active tree structure STRUCT.  If the arrangement of
      alternatives in that tree implies a permutation of alternatives,
      the binary file will NOT carry the alternatives in ascending order
      and great care is necessary to use this data for other than the
      original tree structure.

EXAMPLE:
--------
The above data would be arranged for the binary data set as follows:

1.0 1.0 0.0 1.0  P11 P12 P13 P14  Q11 Q12 Q13 Q14  2  0.5  Y1 S1 A1
1.0 1.0 1.0 1.0  P21 P22 P23 P24  Q21 Q22 Q23 Q24  3  1.5  Y2 S2 A2
etc.,
if tree 12,34 was specified while reading the raw data;

and

1.0 0.0 1.0 1.0  P11 P13 P12 P14  Q11 Q13 Q12 Q14  2  0.5  Y1 S1 A1
1.0 1.0 1.0 1.0  P21 P23 P22 P24  Q21 Q23 Q22 Q24  3  1.5  Y2 S2 A2
etc.,
if tree 13,24 was specified while reading the raw data.

-----------------------------------------------------------------------------
```

3. PROGRAM SEQUENCE, OPTIONS AND OUTPUT

At the beginning, HLOGIT will display the following maximal dimensions which are version specific. Consult the README file included on the distribution diskette.

```
       MAXBUF = size of internal worksheet in memory,
       MAXNP  = total number of parameters,
       MAXTH  = 1st order dissimilarity parameters at level 2,
       MAXTAU = 2nd order dissimilarity parameters at level 1,
       MAXBRA = number of branches on each limb at level 2,
       MAXLIM = number total number of limbs at level 1,
       MAXNX  = alternative specific explanatory variables,
       MAXNXD = agent specific explanatory variable,
       MAXALT = elemental alternatives.
```

A fatal error will occur if one of these maximal dimensions is exceeded.

NOTE: The internal worksheet stores as much of the data file as possible. HLOGIT uses additional memory for computational purposes that are allocated internally and depends essentially on the magnitudes of MAXALT and MAXNP.

HLOGIT will then read the PROFILE. If no profile was specified on the command line and the default profile HLOGIT.PRO does not exist, an error will occur.

In BATCH mode the program will then execute the batch commands listed in the profile starting with line 12. The batch commands correspond to the options listed below.

All batch commands have five character keywords. The following conventions apply:

1. Keywords must begin in colum 1.
2. Keywords may be abbreviated up to minimium identificating length, see examples below
3. The argument list is enclosed in parentheses.

4. Arguments are separated by commas.
5. Arguments must be integers or simple real variables (no exponentials).
6. The argument list can be omitted. In this case, the default values will be used.

```
Examples:
correct:              incorrect:
--------------------  --------------------------------------------------
ESTIM(2,.001,10)      ESTIM(2,.001,10      missing parenthesis
ESTIM( 2 , .001 , 10) E STIM(2,.001,10)    space in keyword
ESTIM()               ESTIM(2,.0 01, 10)   space in number
ESTIM                 ESTIM(2.0,1.E-3,10)  incorrect data types
estim                 ESTIM(2,.001)        too few arguments
ES(2,.001,10)         ESTIM(2 .001 10)     no commas between arguments
es                    E                    too short: same as ELAST
```

In INTERACTIVE mode, a menu will appear and you can choose among the options listed below.

A) ESTIMATION AND STATISTICS:

3.1: ESTIMATION (BATCH: ESTIM / MENU: OPTION=1)

HLOGIT estimates the parameters by full information maximum likelihood.

BATCH SYNTAX: ESTIM(IALG,ACC,ITER) or ES [DEFAULT: ESTIM(2,.001,10)]

INTERACTIVE MODE: If you do not select the default iteration parameters specified in the HLOGIT profile, you will be prompted for:

(1) The numerical optimization algorithm (argument IALG):

1 = Method of steepest ascent.
(Not recommended. Requires no computation of hessian, will always converge, but it converges very slowly. Use only for almost

singular problems.
NOTE: Since the algorithm does not compute a hessian, standard errors and t-statistics are unusable).

2 = Davidon-Fletcher-Powell algorithm.
(Recommended. Will almost always converge, does not require computation of hessian. Will provide estimate of inverse hessian, but this estimate tends to be very poor and is not recommended for covariance estimate)

3 = Newton-Raphson algorithm.
(Requires computation of hessian. Will not necessarily lead to optimum for distant starting values. Use if close to optimum. Recommended for MNL, as in this case convergence is assured)

4 = Berndt-Hall-Hall-Hausman algorithm.
(Recommended. Will always converge, and does not require computation of hessian. Will provide estimate of hessian. In small samples this estimate may be poor)

5 = Quadratic Hillclimbing algorithm.
(Requires computation of hessian. Converges in relatively few iterations, but each iteration very slow. Use if collinearity problems stall other algorithms)

6 = Quadratic Hillclimbing algorithm, hessian updated only every second iteration.
(See 5, needs more iterations, but is more economical per average iteration. On balance sometimes faster, sometimes slower than 5)

(2) The accuracy for convergence criteria (argument ACC):

There are three convergence criteria. Whichever is satisfied first, determines convergence.

1 = Relative change in parameter values between two iterations is less than given accuracy (may indicate a stationary point only).

2 = Norm of gradient is less than given accuracy (convergence assured)

3 = Relative change in function values between four iterations is less than given accuracy (may indicate a stationary point only).

(2) The iteration limit (argument ITER)

This is the inner loop of the optimization program. If this number is exceeded or an exit condition (convergence, failure) is reached, the optimizer returns to the outer loop asking whether additional iterations are desired and whether a new specification of the iteration parameters is desired.

The iteration log includes iteration number and algorithm, the value of the loglikelihood function, the norm of the gradient, the inner product of gradient times whatever is used as direction vector, and the current parameter vector.

NOTE: A small gradient times direction can be used as a sensible convergence criterion only in Newton-Raphson and BHHH iterations.
It is meaningless in steepest ascent iterations. In DFP, the approximation to the hessian (and therefore gradient times direction) is bad unless the number of iterations is close to or exceeds the number of parameters to be estimated.

IN INTERACTIVE MODE: If converged or aborted, HLOGIT prompts you for more iterations, possibly with a different algorithm. If you choose to quit, you will be prompted for the output file name "OUTPARMS.PAR" that creates a file with the estimation output that can be used for later inputs.

IN BATCH MODE: Results are written to the file specified in the profile.

NOTE: It is recommended to employ DFP or BHHH until convergence, then recalculate the covariance matrix by using option 2, below, with analytic derivatives.

NOTE: The output file includes (in parentheses) the following return codes:
-9 = Hessian singular
-8 = Eigenvalues did not converge
-7 = Numerical saddlepoint
-6 = Cannot find improving step
-5 = Too many function errors
-4 = *** Out of memory ***
-3 = Function error in gradient evaluation
-2 = Initial values not admissible

-1 = Iteration limit exceeded

0 = *** Unknown error ***

1 = Stepsize convergence achieved

2 = Gradient convergence achieved

3 = Function value convergence achieved

4 = Gradient*direction convergence achieved

3.2: COVARIANCE MATRIX (BATCH: COVAR / MENU: OPTION=2)

This option lets you re-calculate the covariance matrix, e.g., with a different method of computing second derivatives, and/or prints on screen (punches on file) the entire covariance matrix of the parameter estimates.

NOTE: If the sample is choice-based, the covariance matrix must be corrected in this step.

BATCH SYNTAX: COVAR(IALG,RELSIZ,IPRINT,IPUNCH) or C [DEFAULT: COVAR(3,0,0,0)]

INTERACTIVE MODE: If you do not select the default input parameters, you will be prompted for:

(1) Algorithm (argument IALG)

Algorithm for calculating the hessian on which the covariance is based

0 = no recalculation of covariance matrix

3 = recalculation with exact hessian

4 = recalculation with BHHH approximation of hessian

(2) WESML Correction (argument RELSIZ)

0.0 = simple random sampling

1.0 = choice-based sampling with exact WESML weights

NOTE: A real number different from 0.0 or 1.0 can be used as input for RELSIZ, if the WESML weights are not exact but estimated from some other external ("auxiliary") sample. In this case, RELSIZ must be the ratio of number of observations in the auxiliary sample divided by the number of observations in the estimation sample.

(3) Print Covariance Matrix on screen (argument IPRINT)
0 = no, 1 = yes

(4) Punch Covariance Matrix on result output file (argument IPUNCH)
0 = no, 1 = yes

3.3: PREDICTION (BATCH: PREDI / MENU: OPTION=3)

This option computes a prediction success table in which observed choices are compared to predicted choices. (Predictions are made according to maximum probability).

Likelihood at zero parameters is related to likelihood at estimated parameters (McFaddens Rho: 1-LIK/LIK0), the likelihood and the implied utility (inclusive value at stem level, sometimes referred to as social surplus) at predicted choices is computed.

Aggregate relative choice frequencies (market shares) are computed according to individual maximum probability prediction as well as by averaging over individual choice probabilities.

The choice probabilities and the label (1-9,0,A-Z) of the predicted choice (maximum utility) for each observation can be punched on file.

All statistics take weighting of observations into account.

BATCH SYNTAX: PREDI(IPUNCH) or PRE [DEFAULT: PREDI(0)]

INTERACTIVE MODE: You will be prompted for

(1) Punch predicted choices and probs on output file (argument IPUNCH)
0 = no , 1 = yes

3.4: ELASTICITIES (BATCH: ELAST / MENU: OPTION=4)

This option computes the set of matrices of choice probability elasticities with respect to all exogenous variables E(i,k,j): the elasticity of choice probability i with respect to how variable k is set in alternative j.

Two methods to compute these elasticities are different due to the non-linearity of the NMNL model:

(1) The elasticity is evaluated at sample means and frequencies (quick, but aggregation-biased)

(2) The elasticities are evaluated at each observation and then averaged (slower, but consistent)

All elasticities take weighting of observations into account.

NOTE: If E(i,k,j)=0 for all choice probabilities i and all observations, this row of elasticities is not printed out. For example, this is the case for elasticities referring to explanatory variables that are zero, in particular for the zero case of variables interacted with dummy variables.

BATCH SYNTAX: ELAST(IMODE) or EL [DEFAULT: ELAST(0)]

INTERACTIVE MODE: You will be prompted for

(1) Computation of elasticities (argument IMODE)
    0 = evaluation only at sample means and frequencies
    1 = average of evaluation at each observation
    2 = both

3.5: TREE STRUCTURE TEST STATISTICS (BATCH: TESTS / MENU: OPTION=5)

The significance of the tree-structured NMNL model versus the unstructured MNL model can be tested by comparing the likelihood values. This option computes the other two (asymptotically equivalent) statistics of the test trinity: Wald and Lagrange statistic. Null hypothesis is MNL (i.e., all dissimilarity parameters are one). Degrees of freedom are displayed for the chi-squared statistics.

NOTE: This option yields sensible results only if either preceeded by a successful estimation with reliable standard errors (Option 1, ESTIM) or preceeded by the recalculation of the covariance matrix (Option 2, COVAR). Also, the proper initial values must be present depending on the test, see below.

```
BATCH SYNTAX:  TESTS(ITEST) or T [DEFAULT: TESTS(0)]
```

INTERACTIVE MODE: You will be prompted for

(1) Test (argument ITEST)
    0 = Wald test [DEFAULT]
    1 = Lagrange multiplier test

It is important to make sure that the initial values are correctly specified:

Wald test:
The initial values must be NMNL estimates with estimated dissimilarity parameters. The tree structure must be active, i.e., include commas and/or semicolons.
The covariance matrix must be present. This is the case AFTER successful NMNL estimation.

Lagrange multiplier test:
The initial values must be MNL estimates with unity dissimilarity parameters added to the parameter input file. The tree structure must be active, i.e., include commas and/or semicolons.
The covariance matrix must be present. This is the case BEFORE NMNL estimation, but after a COVAR call.

EXAMPLE:

Assume an IMPARMS parameter file with MNL estimates. Activate the tree structure, i.e., include commas and/or semicolons as appropriate. Append the corresponding number of dissimilarity parameters with unit starting values. Then running the following commands:

```
COVAR
TEST(1)
ESTIM
TEST(0)
```

will produce the correct Lagrange multiplier and Wald tests, provided that the estimation step was successful. Of course, subtracting the initial loglikelihood from the loglikelihood after estimation, and multiplying this difference by two, yields the usual likelihood ratio statistic.

3.6: IMPLIED CORRELATION MATRIX (BATCH: ICORR / MENU: OPTION=6)
===

Each NMNL model (tree structure) implies a covariance matrix of the unobserved utility components in the random utility maximization model where the correlations depend in an factor-like way on common limbs and common branches in the tree. This option computes the full covariance matrix of the unobserved utility components, and normalizes it by taking differences with respect to the last alternative and by scaling the covariance matrix so that the lower right variance is unity. In general, this variance is not unity (e.g., /3 for MNL). The scaling factor is the square root of the lower right variance of the pre-scaled normalized covariance matrix. All parameter estimates should be multiplied by this factor to be comparable to probit estimates with unit variance.

BATCH SYNTAX: ICORR() or I

INTERACTIVE MODE: No prompt.

3.7: SIMULATION OF FINITE CHANGES (BATCH: SIMUL / MENU: OPTION=7)
===

Simulates finite changes of one agent-specific variable while all other variables are held constant. Two sets of choice probabilities and their differences are evaluated at two user-specified values of the variable in question and contrasted with the choice probabilities at sample means. This option provides a discrete alternative to elasticities (OPTION=4) which are often meaningless in the case of dummy variables.

BATCH SYNTAX: SIMUL(IXVAR,IMODE,VAL1,VAL2) or S [DEFAULT: SIMUL(1,2,0,1)]

INTERACTIVE MODE: You will be prompted for

```
(1) Index of variable to be changed (argument IXVAR)
    This index must be between 1 and NXD.

(2) Change and evaluation mode (argument IMODE)

    1 = change evaluated AT MEAN of all variables except changed variable,
        change RELATIVE to mean of changed variable,

    2 = change evaluated AT MEAN of all variables except changed variable,
        ABSOLUTE change between two values specified below

    3 = AVERAGE INDIVIDUAL change after evaluation of all observations at
        original value of all variables except changed variable,
        change RELATIVE to mean of changed variable

    4 = AVERAGE INDIVIDUAL change after evaluation of all observations at
        original value of all variables except changed variable,
        ABSOLUTE change between two values specified below

(3) First value of variable to be changed (argument VAL1)
    If RELATIVE change, this is the fraction
       - of the average value of the variable to be changed
         (if change evaluated at mean)
       - of each observations’s value of the variable to be changed
         (if average individual change)
    If ABSOLUTE change, this is the absolute value to be inserted
       - as average value of the variable to be changed
         (if change evaluated at mean)
       - for each observation as value of the variable to be changed
         (if average individual change)
```

```
(4) Second value of variable to be changed (argument VAL2)
    Same as VAL1 above for the second evaluation

Examples:  SIMUL(1,1,0.95,1.05)  will compute:
           - choice probabilities evaluated at sample means except agent-spec.
             variable No.1, which is set to 5 percent below the sample mean,
           - choice probabilities evaluated at sample means except agent-spec.
             variable No.1, which is set to 5 percent above the sample mean,
           - the differences between these two sets of choice probabilities.

           SIMUL(2,4,0.0,1.0)  will compute:
           - average choice probabilities at sample variables except agent-
             spec. variable No.2, which is set to 0.0 at all observations
           - average choice probabilities at sample variables except agent-
             spec. variable No.2, which is set to 1.0 at all observations
           - the differences between these two sets of choice probabilities.

           SIMUL() will compute:
           - choice probabilities evaluated at sample means except agent-spec.
             variable No.1, which is set to 0.0
           - choice probabilities evaluated at sample means except agent-spec.
             variable No.1, which is set to 1.0
           - the differences between these two sets of choice probabilities.
```

B) UTILITIES
============

3.11: PRINT DATA (BATCH: PRINT / MENU: OPTION=11)
===

Allows you to check data observation by observation. Displays data in expanded form, i.e., with all interactions applied. See the example at the end of Subsection 2.3.3.

BATCH SYNTAX: PRINT(IOBS1,IOBS2) or PRI [DEFAULT: PRINT(1,1)]

INTERACTIVE MODE: You will be prompted for

(1) First observation to be printed (argument IOBS1)
(2) Last observation to be printed (argument IOBS2)

3.12: SPOOL STATUS (MENU: OPTION=12)

====================================

Turns spooling of output on file on/off, and changes the length of the pause between screens of terminal output. Note that specifying a length longer than 60 seconds will turn on a prompt between screens of output.

BATCH SYNTAX: This command is not available in batch mode.

INTERACTIVE MODE: You will be prompted for

```
(1) Spool Status
    0 = no spooling of output
    1 = spooling of terminal output on file

(2) Pause Status
    0 = no pause between screens of output
    1-59 = pause in seconds
    60 = prompt between screens of output
```

3.13: DERIVATIVE CHECK (BATCH: DERIV / MENU: OPTION=13)

===

Compares analytical and numerical derivatives. Should be accurate up to 4 digits unless when close to optimum (zero gradient). If the derivatives do not check, there is an error in the likelihood function and/or derivative routine. Please report.

BATCH SYNTAX: DERIV(IMODE) or D [DEFAULT: PREDI(1)]

INTERACTIVE MODE: You will be prompted for

(1) Comparison of derivatives (argument IMODE)

```
    1 = check of first derivatives only
    3 = also check of exact hessian
    4 = also check of BHHH hessian
```

3.99: QUIT HLOGIT (BATCH: QUITT / MENU: OPTION=99)
===

Quit the program and return to operating system.

```
BATCH SYNTAX:  QUITT() or Q
```

3.20: FILES READ OR WRITTEN BY HLOGIT WITH FORTRAN UNIT NUMBERS
===

INPUT FILES:

```
   "HLOGIT.PRO":   FT18   Profile
   "INPARMS.PAR":  FT20   Tree structure and initialvalues.
   "INDATA.DAT":   FT07   ASCII data file.
   "INDATA.BIN":   FT08   Binary data file.
```

OUTPUT FILES:

```
   "HLOGIT.OUT":   FT10   Spool file in which terminal output is
                          copied (OPTIONAL).

   "OUTPARMS.PAR": FT19   Tree structure and estimation results.
            This file can be used as new tree input file 20.
            Parts A) through C) are copied from "INPARMS.PAR";
            in Part D), the initialvalues are replaced
            by the estimated parameters, incl. standard-errors and t-
            statistics. Appended are summary-statistics.

            NOTE:
```

```
         T-statistics are evaluated for the null-hypothesis
         that the corresponding parameter is 0, EXCEPT for the
         dissimilarity parameters, where the null-hypothesis is that
         the corresponding dissimilarity parameter is 1, the MNL-case.

"COVMAT.OUT":   FT09   Echo of the covariance matrix on file (OPTIONAL).

"PREDIC.OUT":   FT09   Data file of predicted choices (OPTIONAL).

The terminal will prompt you for file names, except for the profile file
which is taken from the command line (or HLOGIT.PRO if omitted). If the
output files already exist, they will be overwritten (after a prompt in
interactive mode, always in batch mode).  If you enter blank names, the
default names from the PROFILE will be used.

-----------------------------------------------------------------------------
```

4. A SIMPLE NMNL TEST PROCEDURE:

Use the following bootstrapping procedure to generate data to test the NMNL estimation results. In the most simple trinary NMNL model

```
  _
 / \
/  /\
```

the following exogenous variables and parameter values:

```
x = (0,0,1) for all observations
 = log2 = 0.6931471806
 = 0.5
```

will create the following choice probabilities:

```
p1 = 0.30902
p2 = 0.13820
p3 = 0.55279.
```

A large data set in which the endogenous variable reflects these choice probabilities will reproduce the above parameter values. For smaller data sets, some bias will occur due to the integer constraint on choice frequencies.

EXAMPLE (n=50 => n1=15, n2=7, n3=28):

```
NMNL ==> lik=-48.05730, a'=0.729822, e'=0.52645
                        (ao=.2747)   (ao=.2213)
                        (t=2.657)   (t=-2.140, measured around 1.0)

MNL  ==> lik=-49.54573, a'=0.93433 , e':=1.0
```

INPARMS file for the above example:

```
full three-level code      quick two-level code      simple MNL-code
---------------------      --------------------      ------------------
12;3                       12,3                      123
1,0,0,0,0                  1,0,0,0,0                 1,0,0,0,0
1   1,1,1                  1   1,1,1                 1   1,1,1
BETA.... 0     0.0         BETA.... 0     0.0        BETA.... 0     0.0
TAU..... 0     1.0         THETA... 0     1.0
```

----

5. MACHINE-SPECIFIC CONSTANTS AND HINTS FOR PORTING HLOGIT

DUPP is largest argument for exponentation
DLOW is smallest number for denominator/logarithm

HARDWARE:	SOFTWARE:	DUPP:	DLOW:
PRIME-MINICOMPUTER	F77	22622.D0	1.D-9824
IBM MAINFRAME	CMS VSFORTRAN	173.D0	1.D-74
IBM-COMPATIBLE PC	MICROSOFT FORTRAN 5.0	708.D0	1.D-307
IBM-COMPATIBLE PC	WATFOR-77/87	708.D0	1.D-307

Some routines are specific for IBM-PC under DOS 3.2 and above.
This is in particular the routine PROFILE which uses a DOS system call to retrieve command line arguments, and SNOOZE and TSTAMP which call the DOS system clock.
The routines PULL and TRANS handle the transfer of data. If the buffer exceeds one memory segment (64KBytes=16000 REAL*4 numbers) the compiler has to use far rather than near addresses in this routine (e.g., by using the $LARGE metacommand).
