HotzTran

User's Manual

By

Robert B. Avery
Board of Governor's of the Federal Reserve System

And

V. Joseph Hotz
UCLA

UNIX Version

First Draft: June 1985
Revised: August 1999

We wish to thank Robert Moffitt and J. S. Butler for providing us with a copy of their routine for estimating a single factor multinomial probit model. We also wish to thank Lars Peter Hansen for his many contributions to the development of this program. Randall Brown also provided useful comments and suggestions. The initial version of the program was written by Avery and has been modified by Hotz.

8/31/99

HotzTran Update
Running HotzTran on a Unix Platform
and
Updates to the Program

1. Running HotzTran on Unix Platforms

For those wanting to use HotzTran on a SSC SUN platform at UCLA—or for those who have downloaded and compiled the source code from the web[1]—you either issue the command:

```
htran.sun
```

for the regular-size version of HotzTran or:

```
htran.sun.big
```

if you need a bigger version for larger data sets. Upon invoking either of these commands, you will see the following:

```
WELCOME TO UNIX(tm) HOTZTRAN

Version as of October 1998

COPYRIGHT (C) 1985, 1998 by Robert Avery and V. Joseph Hotz

Enter the file status (old or new) and file name for each of the files
to be used in response to the following prompts.  Use the format
A3,1X,A40.

Input File =>
```

At the "`Input File =>`" prompt, one needs to enter information about the file containing the INPUT commands that HotzTran will process. In particular, one enters "`old`" because the input file must already exist, followed by the name of the file, such as "`htran.inp`". A copy of a sample version of this input file is provided on the website.

One next sees the following prompt:

```
Data  File =>
```

[1] You can find the source code, User's Manual, and a sample HotzTran run on the website:

http://www.econ.ucla.edu/hotz/.

under "Econometric Software Available."

At the "`Data  File =>`" prompt, one needs to enter information about the file containing the DATA that HotzTran will process. In particular, one enters "`old`" because the data file must already exist followed by the name of the file, such as "`htran.dat`". When first starting with HotzTran, one needs to enter an ASCII or TEXT file containing the data in the FORMAT specified in the INPUT file. A copy of a sample version of the data file that corresponds to the sample input file is provided on the website.

One next sees the following prompt:

```
Print File =>
```

At the "`Print File =>`" prompt, one needs to enter information about the file into which the output of HotzTran will be PRINTED. If this is to be a NEW file, i.e., a file that is not already on created, one enters "`new`". If the file already exists and you wish to write over it, one can enter "`old`". One next enters the name of the file, such as "`htran.pri`". A copy of a sample version of this print file is provided on the website.

One next sees the following prompt:

```
Punch File =>
```

At the "`Punch File =>`" prompt, one needs to enter information about the file into which the intermediate and final set of coefficient or parameter estimates will be printed, or PUNCHED. Again, if this is to be a NEW file, i.e., a file that is not already on created, one enters "`new`". If the file already exists and you wish to write over it, one can enter "`old`". One next enters the name of the PUNCH file, such as "`htran.pun`". A copy of a sample version of this punch file is provided on the website.

One next sees the following prompt:

```
Save  File =>
```

At the "`Save File =>`" prompt, one needs to enter information about the file into which a BINARY version of the data will be SAVED, IF this is requested in the INPUT file. If this is to be a NEW file, i.e., a file that is not already on created, one enters "`new`". If the file already exists and you wish to write over it, one can enter "`old`". One next enters the name of the SAVE file, such as "`htran.sav`". If a SAVE file is created, one can use this file as the DATA file in subsequent runs, so long as one specifies in the INPUT file that the DATA file will be in BINARY format. (See the HotzTran Manual for the appropriate INPUT commands for using BINARY data files.)

Finally, one sees the following prompt:

```
Is the data file formatted (YES/NO)? =>
```

This command is to be answered "`yes`" if one's data file is an ASCII or TEXT file and "`no`" if data file being used is in BINARY format.

After these commands have been entered, HotzTran will execute the INPUT file and create the PRINT, PUNCH and, if requested, the SAVE files.

It is generally easier when running HotzTran to create a file with these responses in a file, "`htran.run`", say, and direct the program to this file via "standard in." For the example being used here, this RUN file would have the following lines in it:

```
old htran.inp
old htran.dat
new htran.pri
new htran.pun
new htran.sav
yes
```

To exploit this way of invoking HotzTran, one issues the following commands:

```
htran.sun.big <htran.run >&htran.log &
```

to invoke a BATCH run, where the output which would normally come to one's screen goes into the file "`htran.log`".

One Final Note: The HotzTran Program is "case-sensitive" when providing it with input commands. The current code is setup so that all input commands documented in the HotzTran Manual should be entered with **LOWERCASE LETTERS** and characters and **NOT UPPERCASE LETTERS** as described in the manual.

2. New Features in HotzTran

2.1 The DOWNHILL SIMPLEX Optimizer in HotzTran:

The SIMPLEX method of Optimization is available in HotzTran. Using this method in HotzTran may be particularly useful when you don't have a good idea about starting values for parameters.

Contained in HotzTran is a new subroutine called amoeba. This routine minimizes the function F0(XSIMP) where XSIMP is the NSIMP x 1 vector of parameters, by a downhill simplex method of Nelder and Mead. It uses, within the routine, the matrix PSIMP whose NSIMP+1 rows are NSIMP-dimensional vectors that are vertices of the starting simplex. [The dimensions of PSIMP are PSIMP(NSIMP+1,NSIMP).] The routine also makes use of the vector YSIMP, which is of length NSIMP+1, whose components must be initialized to the values of F0 evaluated at the NSIMP+1 vertices (rows) of PSIMP. Within the routine, it uses FTOL, the fractional convergence tolerance to be achieved in the function value and the variable STPSIM which is the initial step size used to calculate the SIMPLEX from an initial set of starting values. On output, PSIMP and YSIMP will have been reset to NSIMP+1 new points all within FTOL of a minimum function value, and ITER gives the number of iterations taken.

This algorithm is a modified version of the Downhill Simplex Method algorithm found in "Numerical Recipes," pp. 289-293. It was originally developed by Nelder and Mead, 1965, *Computer Journal*, Vol. 7, p. 308.

To make use of the Simplex Method in estimation within HotzTran, several new options have been created. All of these options are to be used on the MASTER CONTROL CARD in a HotzTran INPUT FILE. The following are new options:

estima = 8 or 9 (Additional values for the option estima)

> While the *estima* is an existing option, it now has two new options. If estima = 8, the program will ONLY use the Downhill Simplex Optimizer. If estima = 9, the program will first use the Downhill Simplex Optimizer and then automatically go into the Fletcher Powell Optimizer. The latter option thus will enable one, in one run, to use the Simplex Optimizer to get one good starting values and then continue with a gradient-based method while the former option just uses the Simplex Optimizer to get good starting values and then stops.

slamb = number (entered in Fortran real*8 form)

> This number is the initial step size used to construct the simplex. It can be either a positive or negative number. The program constructs a tetrahydron by adding the value of slamb to each of the scaled (if scale=0) or unscaled (if scale=1) coefficients to construct the tetrahydron. It may be useful when trying out different starting values for this method to vary slamb (in absolute value and in sign) to search over the estimation surface. The **DEFAULT** is slamb = 1.0.

ssimp = 0,1, or 2.

This integer controls whether the program writes the intermediate values of the full parameter vector associated with the best vertex on the simplex to the usual punch file. If ssimp = 0, it DOES NOT write out the intermediate values at any iteration. If ssimp = 1, it writes the coefficients at the FINAL Iteration. If ssimp = 2, the program writes out the coefficients at each iteration. The **DEFAULT** is ssimp=0.

gtol = number

Again, this is an existing option on the MASTER CONTROL CARD, namely the tolerance for the size of the largest gradient when using either the Steepest Descent or Fletcher-Powell Methods in HotzTran. When using the Downhill Simplex Method, the program will use the gtol value for FTOL, the tolerance described above. The **DEFAULT** value for the gtol is .00001. When using the Simplex Optimizer, I would set gtol = .001 on the MASTER CONTROL CARD so that you don't bog down in trying to get this Optimizer to get you very close to an optimum.

scale = 0 or 1

This option controls whether HotzTran scales the coefficients according to the starting values or not (scale=0 it scales and and scale=1 it does not). You probably will have better luck with setting scale=0 (the **DEFAULT** value). This is based on initial experiences with this optimizer.

sigma = 0,1,2,3, or 4

This option controls whether the variance of the equation error is estimated as a VARIANCE or as its STANDARD DEVIATION. (See manual for full range of options.) When using the Downhill Simplex Minimizer, YOU MUST ALWAYS USE sigma=0,1 or 3. If the Minimizer tries to set these equation error STANDARD DEVIATIONS to negative values, the program will take their absolute values and proceed. In general, initial experience indicates that this optimizer may have difficulty in estimating equation error standard deviations. The user needs to be cautious when using this method on such models.

rho = 0, or 1

This option controls whether the inter-equation error correlations or covariances are, in fact, estimated as correlations or covariances. If rho=0, they are estimated as correlations and if rho=1, they are estimated as covariances. Always set rho=1, i.e., estimate them as covariances. This is necessary because it is not possible to constrain this optimizer from trying values for correlations outside -1 to 1. It might be advisable to initially ignore the inter-equation error correlations and use this optimizer to get you decent starting values for the other parameters and then use gradient methods (Steepest Descent or Fletcher-Powell) in a subsequent run.

When using this optimizer, the program is currently setup to deal with problems in which the total number of parameters being estimated, both restricted and unrestricted, is less than or equal to 700, and the number of unrestricted parameters is less than or equal to 600. If your problem exceeds this amount, you will have to make the following modifications to the HotzTran code.

(1) In the dimension statement below, set the dimension of iplace( ) to a value exceeding the number of total (restricted and unrestricted) parameters used in the problem.

(2) In the double precision statements below change the 200 and 201's to values, k and k+1, where k exceeds the number of unrestricted parameters in the particular problem. Also set the variable INSIM = k in the code of the subroutines AMOEBA and SHRNK.

2.2 MASTER CONTROL CARD OPTION to Check Hotztran's Analytic First Derivatives

Using the estimation option estima=8 and an additional option, a user can have the program calculate numerical first derivatives along with those which have been programmed within HotzTran. This additional MASTER CONTROL CARD Option is *numer*. This option is particularly useful to check analytic first derivatives provided by users with either of the two options noted above. To check first derivatives, the user specifies the following commands:

numer = 0,1.

This integer controls whether HotzTran will printout a comparison between the program's gradients (which are calculated explicitly) and the gradients calculated NUMERICALLY. This option is to be used to test whether programmed gradients (in User-supplied routines) are correct. If numer=0, HotzTran does NOT do a test of numerical derivatives; If numer=1, HotzTran only does a calculation of the numerical and programmed derivative, prints out both sets of derivatives, and then stops. The DEFAULT setting is numer=0. NOTE that when using numer=1, one should ALWAYS set:

estima=8
itera=1
scal=1
slamb= (small number, e.g., .001 or lower [slamb is the PERCENTAGE perturbation in the parameter used to calculate the numerical derivative.]

CAUTION: If the parameter values one is using to evaluate the gradients are such that the gradients are numerically close to zero (as would be the case if the parameters were those associated with an optimum), one may find that some of the numerical and programmed gradients do not agree. One may have to set *slamb* closer to zero (.000001) to accurately calculate numerical derivatives. Even this may not eliminate the discrepancies, which are do to impression in numerically- evaluated gradients. An alternative strategy is to change the parameter values used to evaluate the gradients slightly so that they are not so close to zero.

3. Known Bugs in Hotztran

1. Currently, the options within HotzTran that run Multinomial and Conditional Logits do not work. If you need to run such models, try the packages SST or LIMDEP. Both are available on the system.

2. The option “start=2” used with “model=probit” whenever one is using a model specification which involves estimating a correlation (intertemporal or across equations) *does not work correctly*. The only starting values that are incorrect are those for such correlations; the other parameters are reasonable. The User will have to get starting get starting values for the correlations (pick values close to but not exactly equal to zero) and use the “start=0” option for now.

3. On the DATA CONTROL CARD(S), always set the “print” option to a value greater than 0, even when you don’t want descriptive statistics. For some reason, the program will not read the data correctly unless it first calculates the descriptive statistics.

Table of Contents

1. INTRODUCTION

HotzTran is a FORTRAN based statistical package which is designed to estimate discrete choice, limited dependent, and linear and nonlinear regression models where the models may consist of a system of one or more equations. One of the main features of the package is that it can be used to estimate models applicable to panel data, as well as models specified for cross-sectional data. In panel data contexts, the package will estimate parameters for models in which equation disturbances have very general serial correlation properties.

The package allows one to estimate equations from a fairly large class of nonlinear functional forms (including implicit functions) by simply supplying the package with the form of the equation. Furthermore, the package allows one to easily impose linear and nonlinear restrictions on parameters both within and across equations as well as between equation coefficients and elements of disturbance covariance matrices.

Finally, the package allows the user several options in empirical analysis. They are: (1) the option to utilize alternative schemes to improve the asymptotic efficiency of estimators, (2) the option to perform hypothesis tests of coefficients, overall model fit and the "exogeneity" of certain variables with respect to equation disturbances, and (3) the option to receive a number of "diagnostics" about the fit of the model and of the iterative optimization procedure used in estimation.

The package consists of 3 components: the main program, which is referred to as **HotzTran** throughout the manual, and two subprograms, **MinReg** and **ConTim** which allow estimation of additional statistical models described below, are nested within the main **HotzTran** package, and utilize the data handling and command language set up in the program. The package can be utilized to estimate the following types of models.

1. Models Estimable within the Main Component of the **Hotztran** Package

   a. Single equation models with cross-sectional data:

      - Regression Models
      - Probit Models
      - Tobit Models (with a single or double bound)
      - Logit Models
      - Truncated Variable Models
      - Nonlinear Regression Models (nonlinear in either the parameters and/or the variables)
      - Nonlinear Implicit Function Models (i.e., $f(x,\beta) = \epsilon$)
      - Weighted estimation where the log-likelihood is weighted by a specified variable
      - Regression, Implicit Function, Probit, Tobit, Logit, or Truncated Dependent Variable Models subject to Linear or Nonlinear Restrictions on Parameters

- Regression, Implicit Function, Probit, Tobit, Logit, or Truncated Dependent Variable Models with complex or unspecified (with regressions) heteroskedastic error structures

b. Multiple Equation Systems with Cross-Sectional Data:

- Mixtures of Regression, Tobit, Logit, Truncated Dependent Variable and Implicit Function Equation Models which may be Linear or Nonlinear in the parameters and/or explanatory variables
- Simultaneous Equation Systems
- Consistent parameter estimation while ignoring inter-equation error covariances yet with correct standard errors
- Estimation of multiple-equation systems with full error covariances or with a single factor loading error structure
- Equations Systems which contain both within and across equation Linear and Nonlinear restrictions on coefficients and/or on elements of disturbance variance-covariance matrix
- Selection Models in which one equation determines the censoring on the other equations

c. Models (either Single or Multiple Equation) with Panel Data:

- Consistent parameter estimation of all model forms cited above for Single and/or Multiple Equation Models with Cross-Sectional Data can be obtained without taking explicit account of inter-equation or inter-time period error covariances, but with correctly estimated parameter covariances
- Fixed effect models
- Random effects models for temporal error covariances
- First order autoregressive models

d. The package can also estimate the following more specialized types of models:

- MIMIC (Multiple Indicator, Multiple Cause) Models in which some or all of the indicators are dichotomous or limited variables
- Instrumental variables (IV) Estimates can be obtained for virtually all model forms cited above for Single and/or Multiple Equation Models with Cross-Sectional Data above, where the instruments can be variables left out of equations or values of variables from other time periods. Estimates can use "optimally weighted" instruments, and all standard error calculations will be correct
- Non-linear instrumental variables (NIV) for all the same models forms

2. Models Estimable within the Subprogram **MinReg** of the **Hotztran** Package

   a. Simple moment-based linear regressions can be estimated via Least Squares)

3. Models Estimable within the Subprogram **ConTim** of the **Hotztran** Package

   a. Logistic and Probit choice models can be estimated including:

      - Unordered multiple choice logistic models including conditional logit and multinomial logit
      - Three-choice unordered probit models
      - Ordered logistic and probit models
      - Poisson Regression models
      - Discrete markov models
      - Such models can be represented as linear function of exogenous variables, i.e., $x'\beta$ or, alternatively, in terms of a wide class of non-linear functions of the x's, i.e., $g(x,\beta)$ in place of $x'\beta$, which are specified by the user

   b. Continuous Time Markov and Semi-Markov (proportional hazard) models can also be estimated including:

      - Two-state failure time models
      - Multiple State Models with separate hazard functions for all possible transitions
      - Three forms of duration dependence (Weibull, Gompertz, and, the Box-Cox flexible form) including interactions between duration dependence parameters and exogenous variables
      - Hazard functions are all allowed to depend upon time-invariant and time-varying exogenous variables
      - Hazard functions are allowed to depend upon an individual specific source of unobserved heterogeneity which is assumed to have a gamma distribution
      - The hazard functions for such models can be represented as linear function of exogenous variables, i.e., $x'\beta$ or, alternatively, in terms of a wide class of non-linear functions of the x's, i.e., $g(x,\beta)$ in place of $x'\beta$, which are specified by the user

The **HotzTran** package contains data handling features typically found with similar software. Raw data can be read in a number of different ways with a full range of internal transformations available. Missing data codes can be used and a variety of printed output is available. Data can also be selected for specific runs in a number of ways including ranges of time periods or cross-sectional units or selection by values of variables. In multiple equation systems the inclusion criteria can differ by equation.

A number of options are available to control the estimation process. The convergence criteria used for models requiring iterative solution can be altered easily. Similarly input or "regression run" starting values can be used for such runs and parameter scaling for convergence accuracy is available. Residuals can be printed as can the covariance matrix of estimated parameters. Over-identifying restrictions can also be tested.

HotzTran is designed to be run batch, yet a free form format allows input files to be set up fairly easily. Print output can be either 132 character or 80 character for terminal monitoring. Instructions to the package are read in with alphanumeric "commands". Most commands have built in defaults so only a minimal number need be explicitly specified for a run. Commands can be entered in virtually any order. Variables specified can be referred to either by name or number. Nonlinear equations, variable transformations, and parameter restrictions are specified in easy to construct forms similar to FORTRAN statements. There is also an error-handling procedure which will catch and label most program errors.

In the next chapter, we briefly discuss the basic model forms and data handling features available in **HotzTran.** In Chapter 3, we describe in detail the commands used to run **HotzTran.** In Chapter 4, we offer a brief discussion of Instrumental Variables (IV) Estimation Techniques available in **HotzTran**, estimation with sample selection, MIMIC, and non-linear regression models. In Chapter 5 we provide a number of examples of the commands for running various specifications. In Chapter 6 we describe the commands for the regression subprogram **MinReg.** Chapter 7 provides an overview of the choice and continuous time models estimable in subprogram **ConTim** and the commands needed to execute them. In Appendix 1, we give a list and description of error messages which are generated by **HotzTran.** Appendix 2 provides a brief description of the numerical minimization procedures used in **HotzTran.** Appendix 3 describes the changes to the package that will have to be made for system adaptation. And Appendix 4 describes how to run **HotzTran** on several different systems (IBM, DEC-20 Tops and VAX/VMS systems).

Because **HotzTran** offers a number of options which may be relevant for only a small subset of applications, first-time users may find the length of the manual an imposing barrier to use. Several suggestions may help. If installing a new version of **HotzTran** users should refer first to Appendix 3 which describes internal changes which may be necessary for program adaptation. Appendix 4 should also be consulted for examples of run JCL and/or command structure. First-time users of systems which are already up and running may find it most useful to first skim the examples of Chapter 5. These illustrate the use of options which will apply to most **Hotztran** runs. Detailed followups of specific options and models can then be sought in Chapters 3 and 2 respectively.

2. AN OVERVIEW OF THE MAIN FEATURES OF HotzTran

In this chapter, we present an overview of the major features of **HotzTran**. We defer until Chapters 6 and 7 discussions of the features of the subprograms, **MinReg** and **ConTim**, respectively. In the first section we briefly present the functional or model forms estimable. This is followed in the next section by a presentation of the available methods of model estimation and specification of equation error structures. We provide a separate discussion of instrumental variables estimation, a particular focus of the program. Finally, we conclude with a discussion of program data handling and output specifications. Throughout this chapter, we refer to program commands used to implement procedures with capital letters.

2.1. Functional Forms

A large set of functional (or equation) forms can be estimated with **HotzTran**. These include most of what have been termed "limited dependent variable" models. These include probits, Tobits, double Tobits, truncated regressions, binary logits, as well as more traditional regression models. Detailed descriptions of these model forms can be found elsewhere. However, in the section that follows we give a brief outline of each model as used in **HotzTran**.

2.1.1. Regression Model

By specifying the command MODEL = REGRES, **HotzTran** can be used to estimate the standard regression model of the form:

$$y_{it} = x_{it}'\beta + \delta_{it} \tag{2.1.1}$$

where y_{it} is the observed dependent variable for individual i as of time period t, x_{it} is a vector of explanatory variables and the δ's are stochastic disturbances. **HotzTran** can also be used to estimate nonlinear regressions of the form:

$$y_{it} = g(x_{it}, \beta) + \delta_{it} \tag{2.1.2}$$

Note that the error term must be additive, but the equation can be nonlinear in the x's (NONLIN) or the β's (RESTRI). Furthermore, it is not necessary to specify a dependent variable allowing the estimation of implicit functions. Nor is it necessary to assume that all of the elements of x_{it} are exogenous variables. With the command MODEL = IMPLICIT equations of the following form can be estimated:

$$g^*(x_{it}, \beta) = \delta_{it} \tag{2.1.3}$$

where $g^*(\ )$ is a nonlinear function of the vector x_{it}, some elements of which are endogenous variables, and the unknown vector β. Assuming that β is identified, the program will estimate this parameter vector via instrumental variables (IV) or, if one assumes that the vector δ is normally distributed, via ML.

When linear regressions are specified, **HotzTran** will use moment matrices in estimation whenever possible. This is not possible with non-linear or implicit function models, which require iterative solution methods summing over individual observations.

In situations where the users is only interested in doing linear regressions for single equations, i.e., not as part of a system of equations, the subprogram **MinReg** can be used. It estimates linear regression models via Least Squares and is particularly convenient when the users is running many models with the same dependent variable but alternative sets of independent variables. See Chapter 6 for a discussion of this subprogram.

2.1.2. Probit Model

The binary probit model (MODEL = PROBIT) can be represented in several different ways. One view is that the conditional expectation of an observed binary (0/1) dependent variable is given by a probit (cumulative normal) transform of a linear function of a set of exogenous x variables. An alternative view, and one adopted throughout this manual, is to postulate the following latent structure:

$$y^*_{it} = x'_{it}\beta + \delta_{it}, \qquad (i = 1, 2, ..., N), \quad (t = 1, 2, ..., T) \tag{2.1.4}$$

where the i script indexes the individual of a cross-section and the t subscript indexes the time period in the available panel data set, x_{it} is a k dimensional vector of observed right-hand variables, δ_{it} is an unobserved disturbance, β is an unknown parameter vector, and y^*_{it} is a continuous dependent variable not directly observed by the econometrician. Note, that this specification assumes a panel data set, since many of the particular features of **HotzTran** focus on panel data, although this specification is not necessary. Although y^*_{it} is not observed, the econometrician does observe a censored (or limited) variable which we denote by y_{it}, where:

$$y_{it} = \begin{array}{l} 1 \text{ if } y^*_{it} \geq 0 \\ 0 \text{ if } y^*_{it} < 0. \end{array} \tag{2.1.5}$$

A number of different assumptions can be made regarding the distribution of the δ's and the x's. It is generally assumed that the δ's are stochastically independent of the x's. Furthermore, since the y^*_{it}'s are unobserved there is generally an arbitrary scaling to the probit equation. A common assumption, and the default for **HotzTran** (although it can be over-ridden by the command SIGMA), is to scale the δ's to a variance of one. Finally, the δ's are assumed to

be multivariate normally distributed with several options available as to their joint distribution (see the next section). Given these assumptions the conditional expectation of the observed zero or one y_{it}'s given x_{it} is:

$$E[y_{it} | x_{it}] = Pr\{\delta_{it} > -x_{it}'\beta_0\} = F(x_{it}'\beta_0) \tag{2.1.6}$$

where F( ) is the cumulative normal distribution function. Note that the conditional expectation can also be viewed as the probability that $y_{it} = 1$. If the δ's are independent over time and across people (i. i. d.) then the sample likelihood that $y_{it} = 1$ is given by $F(x_{it}'\beta)$ and the likelihood that $y_{it} = 0$ is $[1-F(x_{it}'\beta)]$. This model can be also characterized as a nonlinear regression model by noting that:

$$y_{it} = F(x_{it}'\beta_0) + \delta_{it}^* \tag{2.1.7}$$

Finally, the program allows one to consider specifications in which the index function, i.e., equation (2.1.4), is expressed as a nonlinear function of x and β plus an additive disturbance. That is:

$$y_{it}^* = g(x_{it}, \beta) + \delta_{it} \tag{2.1.8}$$

Thus a general class of nonlinear function, $g(x_{it}, \beta)$, can be substituted for the linear function $x_{it}'\beta$.

As detailed in the next section, there are a number of different variations of the basic probit model which are available in **HotzTran**. These particularly pertain to the joint distribution of the δ's and affect the estimation strategy employed. Besides distributional variations, it is also possible to model $x_{it}'\beta$ as non-linear in either the x's or the β's (NONLIN or RESTRI).

2.1.3. Tobit Model

The Tobit model (MODEL = TOBIT) is very similar to the probit model, and indeed can be represented as also based on equations (2.1.4) or (2.1.8). However, the outcome variable, y_{it} that the econometrician observes is of the form:

$$y_{it} = \begin{array}{ll} y_{it}^* & \text{if } y_{it}^* > M_{it} \\ M_{it} & \text{if } y_{it}^* \leq M_{it} \end{array} \tag{2.1.9}$$

where M_{it} is an <u>observable</u> lower masspoint which may vary from observation to observation.

Note that one can also consider this same model with M_{it} being a upper bound masspoint (see DEP). Like the probit model, we maintain the assumption that δ is multivariate normal with mean zero but since the y^*'s are sometimes observed, the probit arbitrary scaling of unitary error variances is generally not necessary. Given these assumptions, the expectation of y_{it} given it is above the masspoint is:

$$E(y_{it} \mid M_{it}-x_{it}'\beta<\delta_{it}) = x_{it}'\beta + \sigma\frac{f([M_{it}-x_{it}'\beta]/\sigma)}{1-F([M_{it}-x_{it}'\beta]/\sigma)} \tag{2.1.10}$$

where σ is the equation error standard deviation, f( ) is the standard normal density function, and F( ) is the cumulative normal distribution function. The probability of observing y_{it} at the lower masspoint is $F([M_{it}-x_{it}'\beta]/\sigma)$ while the probability of observing $y_{it} = y_{it}^*$ is $1-F([M_{it}-x_{it}'\beta]/\sigma)$. If observations are i.i.d., the sample likelihood for a non-masspoint observation is $f([y_{it}-x_{it}'\beta]/\sigma)/\sigma$ and for masspoint observations, the sample likelihood is $F([M_{it}-x_{it}'\beta]/\sigma)$. As in the case of the probit model, a number of different options can be used to estimate the Tobit model, which are detailed in the next section.

Like the linear regression model, **HotzTran** will use cheaper moment matrix calculations when estimating Tobit models. These can be used if y^* is linear in the x's, and involve moment calculations only of the non-masspoint observations.

2.1.4. Double Tobit Model

This model is invoked by the command MODEL = DBLTOB. It is similar to the Tobit model in that the equations (2.1.4) or (2.1.8) continues to hold. Now, however, the outcome variable, y_{it} that the econometrician observes is:

$$y_{it} = \begin{cases} 0 & \text{if } y_{it}^* > 0 \\ y_{it}^* & \text{if } M_{it} \le y_{it}^* \le 0 \\ M_{it} & \text{if } y_{it}^* \le M_{it} \end{cases} \tag{2.1.11}$$

so that there is a double truncation point, one from above and one from below. Note that we can also consider the case where M_{it} is a upper masspoint and 0 is the lower masspoint. Characterization of the expectation for y_{it}^* in the region in which y_{it} is continuous and of the probability of observing y_{it} in that region as well as the probabilities of observations being at the upper or lower masspoints is exactly analogous to the Tobit model outlined above. The expectation of y_{it} given it is not at either masspoint is:

$$E(y_{it} \mid y_{it} = y_{it}^*) = x_{it}'\beta + \sigma \frac{f([M_{it}-x_{it}'\beta]/\sigma) - f([0-x_{it}'\beta]/\sigma)}{F([0-x_{it}'\beta]/\sigma) - F([M_{it}-x_{it}'\beta]/\sigma)} \tag{2.1.12}$$

where σ, f( ), and F( ) are defined as before. The probability that y_{it} will be at the lower masspoint, M_{it}, is $F([M_{it}-x_{it}'\beta]/\sigma)$. The probability that y_{it} will be at the upper masspoint, 0, is $1-F([0-x_{it}'\beta]/\sigma)$, and the probability that y_{it} will be at neither masspoint is one minus the two masspoint probabilities. The sample log-likelihood for the non-masspoint observations is the same as the regular Tobit model, and for masspoint observations is equal to the log of the masspoint probability.

2.1.5. Truncated Dependent Variable Regression Model

In the Truncated Regression Model (MODEL = TRUNCATE), again we use equation (2.1.4) or (2.1.8) but now the outcome variable, y_{it} that the econometrician observes is:

$$y_{it} = y_{it}^* \quad \text{if } y_{it}^* > M_{it} \tag{2.1.13}$$

while if $y_{it}^* < M_{it}$, we do not observe y_{it} at all. (As above we can also consider this same model with M_{it} being a upper bound masspoint.) Identical assumptions are made as with the Tobit model regarding the distribution of the δ's. The expectation of y_{it} given it is in the sample is identical to the conditional expectation of a Tobit dependent variable given it is not at a masspoint:

$$E(y_{it} \mid M_{it}-x_{it}'\beta < \delta_{it}) = x_{it}'\beta + \sigma \frac{f([M_{it}-x_{it}'\beta]/\sigma)}{1-F([M_{it}-x_{it}'\beta]/\sigma)} \tag{2.1.14}$$

where σ, f( ), and F( ) are defined as before, with $1-F([M_{it}-x_{it}'\beta]/\sigma)$ being the conditional probability of the observation being in the sample. If observations are i.i.d., the sample likelihood of an observation is $[f([y_{it}-x_{it}'\beta]/\sigma)/\sigma]/[1-F([M_{it}-x_{it}'\beta]/\sigma)]$. With a few exceptions, estimation of this model proceeds in exactly the same way as that outlined for the Tobit model. There are, though, some limitations on the types of underlying disturbance structures for which consistent estimates of parameters can be generated by the procedures available in **HotzTran**. These are detailed in the next section.

2.1.6. Logit Model

The logit model (MODEL = LOGIT) assumes that δ_{it} in equations (2.1.4) or (2.1.8) has a Weibull or extreme value distribution. Then the conditional expectation that $y_{it} = 1$ is:

$$P_{it} = \frac{1}{1+\exp(-x_{it}'\beta^*)} \qquad (2.1.15)$$

where $\beta^* = \beta/\sigma$ and where the expectation that $y_{it} = 0$ is just $1 - P_{it}$. The likelihood is formed identically to the expectation. Like the probit model, σ is normalized to one and not estimated by **HotzTran** (this cannot be overridden with Logit). As with the Truncated Regression model, there are some limitations on the error distribution assumptions and estimation procedures that can be used with a logit specification.

2.2. Equation Error Structures and Estimation Methods

HotzTran can be used to compute *Maximum Likelihood* (ML) estimators under a number of different assumptions. Each of the model forms mentioned previously (probit, Tobit, logit, truncated, regression, double Tobit) can estimated with ML. The critical consideration in determining the feasibility of ML is the data pattern. Four different data patterns are possible: (1) single-equation/non-panel data; (2) single-equation/panel data (multiple time periods); (3) multiple-equation/non-panel data; and, (4) multiple-equation/panel data. ML models can be estimated for each of these data patterns but different specifications and/or assumptions will be required. In a number of cases, "pseudo-ML" estimators can also be computed which will not be fully efficient but will be consistent and will have correct standard errors. Below we briefly outline the ML (and pseudo-ML) estimators available.

2.2.1. Multiple-Equation/Non-Panel (Single Time Period or Pure Time Series) Data

If it is assumed that observations are independent (as seems reasonable with cross-sectional data) then ML estimates can be computed with option TYPE = 1. These estimates will be the standard single-equation i.i.d. ML estimates normally seen for models like probit etc. For regressions the computed estimates will be least squares estimates. The sample likelihood maximized is the i.i.d. likelihoods given in the previous section on models estimable with **HotzTran**. In all cases ML estimates of both the parameter vector and equation error variances will be computed. Observations need not be identically distributed, as it is possible to have error variances vary by observation. This specification requires the use of the NONLIN = 1 option.

With a pure time series, correct parameter estimates can be computed for most models with TYPE = 1 even if equation errors are correlated. In most instances, however, these estimates will not be ML and the parameter standard errors printed will not be correct without adjustment. The program will compute correct standard errors in these cases if errors have a stationary covariance matrix given by a moving average process of a limited order, t, known to the user. The correct standard errors can be invoked by the commands PSTAN = 1 and MAVER = t.

2.2.2. Single-Equation/Panel Data (Multiple Time Periods)

The critical consideration with panel data is what assumptions are made about the distribution of the disturbance vector, δ, or in some cases δ_i, for a cross-sectional unit, particularly its covariance matrix, Σ.

If one assumes that Σ is diagonal, then the program can generate ML estimates of those elements and of the parameter vector, β, using the TYPE = 1 option (similar estimates can be computed with TYPE = 3 as well). This assumption is equivalent to the assumption that observations are independent over time. However, even if observations are not independent over time, i.e. Σ is not diagonal, option TYPE = 1 can still be used. However, estimators will no longer be ML (see Chapter 4 for a more extensive discussion of this), although they will be consistent. The option PSTAN = 1 will produce correct standard errors of the estimators under general conditions for the error correlations (see MAVER).

Whenever panel data systems are estimated under the assumption that errors are serially uncorrelated (TYPE = 1 or TYPE = 3), it is also possible to compute consistent estimates of the inter-time error covariances (or correlations). This is done with the option TCORR = 1 (or 2 or 3). These estimates are computed by maximizing the conditional likelihood of each element of Σ given the coefficient vector which is previously estimated. These estimates will not be ML since the coefficients and error disturbance structure are estimated sequentially, not concurrently. They will however be consistent and the program will print the correct standard errors.

There are several options available in **HotzTran** which may make the assumption of intertemporal error independence more palatable. The program transformations make it easy to include time dummies for each time period, for example. It is also possible to adjust variables to "deviations about their cross-sectional means" using the TIME VARIATION CARD. Another similar alternative is a *fixed effect* model. This specification is equivalent to adding a separate intercept term for each cross-sectional unit. Alternatively, it can characterized by specifying the disturbance term, δ_{it} as:

$$\delta_{it} = w_i + u_{it} \tag{2.2.1}$$

where w_i is a fixed constant which can be different for each individual. It is implemented by the command FIXED = 1 and can be used for all functional forms with TYPE = 1 or TYPE = 3. Note though, that this option does require variation over time in the dependent variable for probit, logit, Tobit, and double Tobit models. Program standard error calculations will not be quite correct with fixed effect models, as they will not take full account of the correlation between the fixed effects and other independent variables.

ML estimates with panel data can also be computed under the assumption that the error covariance structure can be described by a *random effects* specification. In this structure, we assume that δ_{it} can be written as:

$$\delta_{it} = v_i + u_{it} \tag{2.2.2}$$

where v and u are random disturbances independently and normally distributed with variances σ_v^2 and σ_u^2, respectively. This specification is equivalent to the assumption that the off-diagonal elements of Σ are equal (equi-correlated). This specification can be implemented using the TYPE = 2 and RANDOM = 0 options. The actual estimation procedure used utilizes a hermite integral approximation of the type describe by Butler and Moffitt (1981). It cannot be used with logit or truncated, however. Note that unlike TYPE = 1, these estimators will generally not be consistent unless the error covariance structure truly is equi-correlated.

ML estimators can also be computed with single-equation panel data under the assumption that the serial correlation is characterized by a *first order autoregressive* structure. Such a specification arises under the situation where δ has the following structure:

$$\delta_{it} = \rho \delta_{it-1} + u_{it} \tag{2.2.3}$$

where ρ is a constant parameter taking values between 0 and 1, and the u's continue to be independently and normally distributed. This specification can be implemented using the option TYPE = 2 and RANDOM = 1. It cannot be used with logit or truncated. Like the equi-correlated case, estimates will generally be consistent only when the first order autoregressive assumption is correct. When there are only two time periods, the autoregressive and random effects estimates will be the same and both will always be ML.

2.2.3. Multiple-Equation/Non-Panel Data (Single Time Period or Pure Time Series)

The critical consideration with multiple-equation systems is what assumptions are made about the cross-equation error covariances.

If one assumes that errors are uncorrelated across equations, then the program can generate ML estimates of coefficients and error variances using the option TYPE = 1. This option allows equations to be estimated jointly, hence cross-equation parameter restrictions imposed. Even if equation errors are not independent, option TYPE = 1 can still be used. However, estimators will no longer be ML (see Chapter 4), although they will be consistent. By using the option PSTAN = 1 (and MAVER with a time series), the program will generally also compute correct estimates of the parameter standard errors. Whenever multiple-equation systems are estimated under the assumption of uncorrelated equation errors (TYPE = 1 or TYPE = 3), it is also possible to also compute consistent estimates of the cross-equation error covariances (or correlations). This is done with the option ECORR = 1 (or 2 or 3). These estimates are computed by maximizing the conditional likelihood of each cross-equation error covariance term given the coefficient vectors which are previously estimated. These estimates will not be ML since the coefficients and error disturbance structure are estimated sequentially, not concurrently. They will however be consistent and the program will print the correct standard errors.

ML estimates for two-equation systems with correlated errors can be computed with the option TYPE = 4. ML estimates are computed for the coefficient vectors, both equation error variances, and the equation error covariance (or correlation). These bivariate ML estimates can be computed with any two-equation system except those involving logit or truncated equations.

Even with more than two-equations, the TYPE = 4 option can be used. However, estimates will generally not be ML, although they will be consistent. The program will also compute correct standard errors. Estimates will be computed not only for parameter vectors, but for elements of the cross-equation error covariance structure as well.

ML estimates can also be computed whenever the cross-equation error disturbances can be described by a single *factor analytic* specification. In this case, if the disturbance for individual i's k^{th} equation is δ_{ik}, the factor analytic model implies the following structure on these disturbances:

$$\delta_{ik} = \gamma_k v_i + u_{ik}, \qquad (k=1,...,K) \tag{2.2.4}$$

where γ_k is the factor loading on the single factor v_i for the k^{th} equation. In this specification we assume that the v's are normally distributed, independently of u, and with variance equal to one and that the u's are independently and normally distributed with variance σ_u^2. This specification can be implemented using the option TYPE = 2 and RANDOM = 2. The estimation procedure is identical to that of the random effects model. This option cannot be used with logit or truncated. Note again, that estimates will generally not be consistent unless the factor analytic assumption is correct. It will yield the same estimates as TYPE = 4 when there are only two equations. Finally, note that it generally is necessary to normalize at least one parameter loading (say for equation one) in order to estimate the model.

2.2.4. Multiple-Equation/Panel Data (Multiple Time Periods)

Any specification that can be implemented for panel and/or multiple-equation systems can be implemented when both are present. However, when implementing any complicated inter-temporal ML procedures (random effects or first order autoregressive) it will be necessary to assume that cross-equation errors are uncorrelated for ML (or utilize quasi-ML procedures which ignore the correlations). The program, however, will correctly adjust standard errors to account for the fact that there are multiple equations. Inter-equation error correlations can still be computed (ECORR > 0) averaging over all time-periods.

Similarly, when implementing any complicated inter-equation ML procedures (factor analytic or TYPE = 4) it is necessary to assume that inter-temporal error covariances are zero for ML (or utilize quasi-ML procedures which ignore the correlations). Again, the program can be asked to compute standard errors taking the panel data structure into account (PSTAN = 1). Inter-temporal error correlations can still be computed (TCORR > 0) and will be calculated separately for each equation.

2.2.5. Instrumental Variables Estimation

The user of **HotzTran** has the option of using Instrumental Variables (IV) estimation to estimate many of the models described above. While a more complete discussion of IV estimation (in the context of estimating a multiperiod probit model) is given in Chapter 4, here we briefly indicate the types of options available.

In **HotzTran**, the user has a number of model forms -- probit, Tobit, regression, etc.--to choose from. Each form gives rise to a functional form expressed in terms of the model form's disturbance term or its forecast error. This is the discrepancy between the observed dependent variable and the conditional expectation function for the particular model form. Consider a sample of N individuals and T time periods. Denote these disturbances or forecast errors as δ_{it} for the i^{th} individual in the t^{th} time period. This disturbance arises from the model just described and is a function of the individual's y's and x's and the model's unknown parameters, β, i.e., $\delta(y_{it},x_{it},\beta)$. Denoting the i^{th} individual's vector of disturbances as $\delta(y,x,\beta)' = (\delta(y_{i1},x_{i1},\beta),....,\delta(y_{iT},x_{iT},\beta))'$, then IV estimation is based on the fact that there exists a matrix, $Z_i = (z_{i1},....,z_{iT})$ such that:

$$E(Z_i \delta(y,x,\beta)) = 0 \tag{2.2.5}$$

where z_{it} is a vector of *instruments* for the i^{th} individual that are to be used in the t^{th} time period.[1] Since $\delta(y,x,\beta)$ is a function of parameters, β, say, IV estimation proceeds by setting sample analogues of expression (2.2.5) equal to zero. The sample *orthogonality conditions* are:

$$M(\beta,y,x,z) = \sum_{i=1}^{I} \sum_{t=1}^{T} Z_i \delta(y,x,\beta)) \tag{2.2.6}$$

Then IV estimation is implemented by finding estimators of β, b_N, so as to minimize:

$$O(b_N) = M(b_N,y,x,z)' W\ M(b_N,y,x,z) \tag{2.2.7}$$

where W is a *weighting matrix*. Such estimators are consistent and asymptotically normal with a covariance matrix given by:

$$(D_0'WD_0)^{-1}(D_0'WS_0WD_0)(D_0'WD_0)^{-1}, \tag{2.2.8}$$

where

$$D_0 = E\left[\frac{\partial M(\beta,y,x,z)}{\partial \beta}\right]$$

---

[1]The instruments can actually be variables which come from other time periods. For example, in certain models x variables from past time periods can serve as instruments in time period t. Implementing the use of instruments for period t which come from other time periods can be accomplished using the LAGS option.

$$S_0 = E[M(\beta,y,x,z)M(\beta,y,x,z)'].$$

Note that the above covariance matrix is quite general; in particular it allows the δ's to be conditional *heteroskedastic* where the form of heteroskedasticity is unknown. Thus estimation can proceed under the general situations considered in the recent econometrics literature by Hal White, for example. W can be either the identity matrix or an optimally constructed weighting matrix. The program will construct the latter as a function of the y's, x's, z's and consistent estimators of β.[2] The program will provide the estimators b_N as well as consistent estimators of its covariance matrix. S_0 can also be estimated under several different assumptions.

To implement IV estimation in **HotzTran**, the user must set the TYPE option on the MASTER CONTROL CARD equal to 3 (TYPE=3) and can choose:

1. the form of $\delta(y_{it},x_{it},\beta)$, using the MODEL options on either the MASTER CONTROL CARD and BASIC EQUATION CARD.

2. the elements of z_{it}, i.e., the instruments, on the EQUATION VARIABLE CARD and using the LAGS option.

3. the weighting matrix, W, using the OPTWT option.

4. the S_0 matrix estimator, using the MAVER option.

Note that **HotzTran** can deal with nonlinear forms of the δ function.

2.3. Data Handling and Estimation Controls

In this section we briefly discuss the non-statistical features of **HotzTran**. We begin with an overview of data and file handling and variable transformations. We follow with a discussion of data selection for particular models or equations. Finally we conclude with a discussion of estimation and output controls.

2.3.1. Data Input and Transformations

HotzTran has a full range of data handling features available which are comparable to those of similar software packages. These include a variety of different ways to input and store data, missing data codes, a full range of transformation codes, and print options. Detailed options include the following.

HotzTran is designed to input a primary observation file which it processes, adds to by variable transforms, and then stores either on disk or in core. A series of models or "runs" can then be estimated by selecting subsets of observations and/or variables specified on

---

[2]The form and construction of Optimal Weighting Matrices are discussed in Chapter 4. They are implemented in **HotzTran** with the MASTER CONTROL CARD option OPTWT described on page 36.

separate control cards. As many runs as desired can be run utilizing the same primary observation file. The program will create a separate observation file for each run using only needed observations or variables, which it also stores on disk or in core.

Because panel data is so important to the models used in **HotzTran**, the primary observation file can have two dimensions -- cross-section or individual units (OBSV) and time periods (TIME). Observations must be read in sequential order one cross-section unit after another, and by ascending time period within cross-section unit. Time period numbers, however, can be determined by the value of an input variable (TIME = V#). This allows cross-section units to have a varying number of time periods. Pure time series or cross-section samples can also be used.

Observations can be read by a variety of means. They can be read in formatted form by an input format (FORMAT = (...)); binary read (FORMAT = 0); or utilizing a user supplied FORTRAN program. Two such routines are available. SUBROUTINE USER is called once for each observation read. The subroutine is fed the number of the observation, the input device number, and an observation vector to fill. The subroutine can be used to handle complicated reads or transformations which are simpler with user-supplied FORTRAN code. SUBROUTINE USEO can be used in a similar fashion, but instead of one observation at a time, the routine is used to return the entire data matrix.

Several other features on data input are available. A missing data code (MISS) can be used to flag missing data on a variable by variable basis (the code must be the same though for all variables). Observations can be read until an end-of-file is encountered (or signaled by a return from SUBROUTINE USER). The primary observation file can be stored in core or on disk (DISK). The input file device can also be set (INPUT).

Several print options are also available. The individual observations can be printed as well as the sample means, standard deviations, maximums and minimums, correlation matrix, and moment matrix (PRINT). The print line width can also be set at 132 or 80 characters to accommodate terminal viewing (LINE). Variables can be named with up to 8 character names or referred to by number (NAME).

Finally, a full range of variable transformations are available. Transformations can be set to loop over a range of observations/time periods/variables similar to a "do loop" (FOR(# TO #)). Transformations can also be controlled by "IF" statements. These can depend on the value of the time period (IFTIME), the cross-sectional unit number (IFCROS), or a variable value (IF( )).

Transformations can involve existing variables or create new ones. Variables can be referred to by name or by number. The transformations allow most "FORTRAN" type expressions using +, -, *, /, **, and log and exponentiation. A single level of parenthesis can be used as well. Special features are available to capitalize on the time dimension of data. A time trend (VT0) or time dummy variables (VT#) can be used in transformations. Variables can also be led or lagged automatically by referring to them with lead or lagged subscripts. The program will set transformed variables dependent on missing data automatically to the missing data code as well.

2.3.2. Observation Selection for Model Runs

A number of options are available in **HotzTran** to select observations and data for a particular run. Codes are also available to allow separate time periods and/or cross-sectional units to be used on an equation-by-equation basis. These criteria can be changed from run to run.

Variables with missing data codes can be handled in variety of ways for specific runs. In particular, one can choose to include all observations even those with missing data (MISS = 0), to exclude any observations from the entire estimation if any independent or dependent variables are missing anywhere in the system of equations being estimated (MISS = 2) or to exclude observations only from the computation of the estimators for the particular equations in which the variables are missing (MISS = 1). In all of these procedures, the program properly computes the estimators of interest and properly computes the standard errors and test statistics associated with such estimators.

The user can also control the data coming from particular observations or time periods which are used in the estimation or systems of equations or particular equations. The options UTIME, XTIME, UCROS, and XCROS on the MASTER CONTROL CARD allow the user to indicate the range of time periods and cross sectional observations to be used in the estimation of all the equations to be estimated in a particular run. Furthermore, the option UOBSV allows the user to exclude certain observations based on some variable taking on certain values. (For example, this option could be used to estimate an equation only for individuals who were working where a variable taking on a value of 1 if the individual was working and 0 otherwise was coded and included in the dataset.) The user also can control which observations are used in the estimation of a particular equation within a system of equations. This is accomplished using similar equation-specific options -- UCROS, XCROS, UOBSV -- on the BASIC EQUATION CARD. This will determine which observations will be used in the estimation of a particular equation. Note that these features mean that the program is flexible enough to compute estimators in which different number of observations may be used in the estimation of parameters in different equations within a system of equations. Again in utilizing any of these options, the program properly computes the estimators of interest and properly computes the standard errors and test statistics associated with such estimators.

Several other data handling features are also available for particular runs. The sample log-likelihood can be weighted by the value of an input variable (WEIGHT). Run observations can be stored in core or on disk (DISK). Data can be utilized as deviations about their cross-sectional means (TIME VARIATION CARD) or stored parsimoniously if they don't vary over time (TIME VARIATION CARD).

The specification of variables used in models can also be done in several different ways. Variables can be referred to by names or by numbers (V#). Equations can be given by variable ranges such as V1-V23. Variables can be excluded from such ranges by NV# specification. A constant can also be added automatically to equations (CONST) or used selectively. Dependent variables for limited dependent variables can be given in easy to specify form. For example, DEP=V3.GT.2. in a Tobit model would state that the dependent variable was V3 with a lower masspoint of 2.

Non-linear equations (NONLIN) and parameter restrictions (RESTRI) can also be specified in a simple form. Non-linear equations are written as functions of parameters (B's) and variables (V's or they can also be named.) Most FORTRAN type operations such as +, -, *, /,

**, and log can be used. A single level of parenthesis is also allowed. These allow most all non-linear models to be specified. Just equations can be specified or equation variances and covariances made non-linear as well. The program will compute the correct analytical first and second derivatives of the non-linear equations for use in iterations and standard error calculations.

Parameter restrictions can be imposed in estimation using a form similar to NONLIN. Restricted parameters are set equal to functions of unrestricted parameters. These can be linear or non-linear functions, again using +, -, *, /, **, and Log and a single level of parenthesis. Correct analytical first and second derivatives (hence standard errors) will be computed.

2.3.3. Estimation and Output

Most models in **HotzTran** are designed to be computed by iterative solution methods. This is true of all the limited dependent variable models, as well as instrumental variable models, non-linear models, FIML multiple-equation models, and models with non-linear parameter restrictions. Thus control of the iterative solution process is important. A number of controls are available which are detailed in Appendix II. Briefly, these include the following.

The program uses a Davidon-Fletcher-Powell iterative solution scheme which is detailed in Appendix II. Iteration starts using only the function gradient (STEEPEST DESCENT) and after reaching a measure of convergence, gradients weighted by second derivatives (FLETCHER-POWELL). The program does not compute second derivatives at each iteration, but modifies a matrix. The choice of estimation methods can be set (ESTIMA), as well as the convergence criteria: function change, gradient size, and parameter change (set by CNVRG). The tolerance of convergence can also be set (GTOL, FTOL, and PTOL). It is also possible to control the number of iterations (ITERA) and CPU seconds used (SECS). If outlier observations are encountered during estimation they can be handled in several different ways by OUTLY.

Iteration success can be very sensitive to scale. A scaling provision is available in **HotzTran** to scale both the data and parameters for best ease in convergence (SCALE = 0). Starting values can also be set in various ways. They can be read in or computed from least square regressions (see START). The starting regressions are adjusted to take account of parameter restrictions, fixed effects, random effects, or first order autoregression. If starting values are read in they can use a preset or read in format (FORMAT).

Finally, a number of different print and punch options are available for individual model runs. Print output can have 132 or 80 character lines (LINE). A title card is used on all output (see RUN TITLE CARD). Parameter standard errors can be computed and printed in three different ways (PSTAN). The parameter covariance matrix can be printed (PCOVA). Iteration information can be printed (IPRINT). In addition to standard measures of fit, residual and predicted value analysis can be computed and printed for the observation run file (RPRINT).

Three options are available to punch data (write on disk). Parameter estimates can be punched at the conclusion of each run (CWRITE = 1), or at each iteration (CWRITE = 2). Residuals can be punched (RWRITE) or the estimated covariance matrix of the errors of a multiple-equation system (MWRITE).

3. COMMANDS FOR RUNNING HotzTran

3.1. Introduction

Each of the commands (individual instructions) on each of the following "CONTROL CARDS" or "CONTROL LINES" can be typed in free format with multiple commands per line. Commands should be separated by blanks or commas. Blanks should generally be avoided within commands, but can be used around ='s or within equations. Only the first two letters of a command are necessary (e.g. VARI or VA), but *ALL* COMMANDS must be entered in *UPPERCASE LETTERS*. Control cards can extend more than one line (a line is at most 80 characters). If a "card" extends more than one line, place a $ at the end of each line that is extended. Line breaks should not be in the middle of a command, but can be in the middle of equations. Defaults are shown for most commands. If the command is not specified, then it will assume the default value. The options can be placed in any order on the particular control card.

The DATA CONTROL CARD is read once for each data file used (sequentially). This control card determines the structure of the input data file, variables, and any variable transformations. Data can be read in panel or straight observation form (if panel data, all time periods do not have to be read in for each cross-sectional unit.) Data can be read from a binary, formatted, or user-supplied subroutine mode. Various print options of the data are also available. Run printout can be with 132 or 80 character lines for monitoring from a terminal. An option is also available to use a missing data code. The RUN TITLE CARD, the VARIABLE NAMES CARD(S) (optional), and the DATA TRANSFORMATION CARD(S) (optional) are each also read just once following the DATA CONTROL CARD.

The MASTER CONTROL CARD, TIME VARIATION CARD (optional), BASIC EQUATION CARD, EQUATION VARIABLE CARD (optional), EQUATION TESTING VARIABLE CARD (optional), ORTHOGONALITY CONDITION TEST CARD(S) (optional), NON-LINEAR EQUATION CARD (optional), RESTRICTION CARD (optional), FORMAT CARD (optional), COEFFICIENT STARTING VALUE CARD (optional), WEIGHTING MATRIX COEFFICIENT CARD (optional), TIME PERIOD CORRELATION CARD (optional), and EQUATION CORRELATION CARD (optional) are repeated (in order) for each "run" or "model". Many of these control cards are optional depending on the model and techniques desired. As many models as desired can be run. When no more model runs are desired, just type STOP on the next MASTER CONTROL CARD. The program can also return to the beginning and read another DATA CONTROL CARD and data file, by typing the command SAMPLE on a TRANSITION CARD placed after other cards for a run. It is also possible to run regressions in subprogram **MinReg** or logistic choice models or continuous time markov models in subprogram **ConTim** instead of stopping or reading another sample by using the MASTER CONTROL CARD (**MinReg**) or (**ConTim**) (see the chapters on these subprograms).

Many of the detailed options are best explained in the write-up of the individual options which follows. The basic model specifications are determined on the MASTER CONTROL CARD. Each model can be single or multiple equation. Models can be estimated assuming a panel data structure or not. Different subsets of the observations and/or time periods can be used. Within a model equation system, each equation can have a logit, probit, Tobit, truncated, double (upper and lower masspoints) Tobit, or regression dependent variable. Each model system can be estimated in different ways depending on assumptions about the inter-equation or inter-time period error correlations. Options are also available for non-linear restrictions in

the parameters and/or non-linear equations. Several orthogonality condition testing procedures are also available. All models are estimated using an iterative Davidon-Fletcher-Powell algorithm (described later). A number of different convergence, scaling, and tolerance levels can be adjusted. Options are also available to alter the print-out and do residual analysis. Parameters can also be punched (written on disk). The log-likelihood can also be weighted by a specified input variable.

The specification of the individual equations of each "model" is done on the BASIC EQUATION and EQUATION VARIABLE CARD which follow the MASTER CONTROL CARD for each model equation. Both independent and dependent variables are selected from the main observation file. They can be referred to either by their number in the main observation vector, or by their names as read in on the VARIABLE NAMES CARD(S). A constant is referred to as variable 0 or CONSTANT. Variables can be used as normal independent variables; however, they can also be specified as instruments or endogenous. Lead or lag values of the variables can also be used as instruments. With panel data, equations can be estimated with fixed or random effects or with a first order autoregressive error.

Since the program uses iterative procedures, starting values for the parameters are important. These can be generated by the program itself (from regressions) or read on the COEFFICIENT STARTING VALUE CARD. The user may supply the format used to read the starting values as well.

The TRANSITION CARD signals the end of the main **HotzTran** runs and will command the program to read a new data file and DATA CONTROL CARD (and other setup cards) with the command SAMPLE. It can also be used to rewind or end file data files.

The details of the control cards and individual command instructions are given in the text which follows. Most options have system defaults which apply when they are not explicitly specified by the user. If an option does not seem to pertain to the model you are running, then ignore it. A number of examples of program runs and sample card decks are given in Chapter 5.

3.2. DATA CONTROL CARD

The following options apply to the structuring of the input data, including data transformations, and to the choice of summary statistics printed out by the program.

Option	Description
VARI = #	where # is any integer > 0. There is no default, and VARI must be specified. VARI specifies the total number of variables per observation that will be read from the input data file. This only includes input variables and does not include any variables created on the DATA TRANSFORMATION CARD(S) nor the constant. Thus, the total number of usable variables including transformations, VAR*, may be larger than VARI. A constant (referred to as variable zero) will always be added at the end of the list of variables (it is not included in VARI or VAR*). Thus a constant need not be provided by the user.
OBSV = #	where # is any non-negative integer. The default is 0. OBSV is generally the total number of cross-sectional units used with panel data. If non-panel

data is used, or time periods are variable (see TIME), OBSV is the total number of observations. If OBSV = 0 (default) the program will read to end-of-file (if FORMAT is binary or read in) or will read until IDD = 2 or IDD = 3 (if SUBROUTINE USER is used) or will be set by the return from SUBROUTINE USEO. If fewer than OBSV observations are actually read, the program will use the smaller amount.

TIME = $\#_1$ or $V\#_2$

where $\#_1$ is any integer > 0 and $V\#_2$ is any integer from 1 to VARI. The default is $\#_1$ equal to 1. If $\#_1$ is specified TIME is the total number of time periods (applies only to panel data). The total number of observations is OBSV*TIME. Observations must then be read for each time period of each cross-section unit. Observations are then assumed to be ordered by time, and read as:

cross-section unit 1, time period 1;
cross-section unit 1, time period 2;
.
.
cross-section unit 1, time period TIME;
cross-section unit 2, time period 1;
.
.
cross-section unit OBSV, time period 1;
.
.
cross-section unit OBSV, time period TIME.

If $V\#_2$ is used, the time period of each observation is input as the value in variable $V\#_2$ (variable names cannot be used). Thus, missing time periods are allowed. Observations, however, must still be read in ascending order by time period. The program assumes a new cross-sectional unit whenever the time period of an observation is lower (no greater) than the time period of the previous observation, or the specified time period is negative (the observation is assigned its positive value for a time period). If $V\#_2$ is used then OBSV is the total number of observations read. If $V\#_2$ is used, and DISK = 1 or the maximum number of time periods is greater than 10, then the maximum number of time periods per cross-section unit must be specified. This is done by also specifying TIME = $\#_1$ on the DATA CONTROL CARD, where $\#_1$ is the maximum number of time periods. Thus, TIME = V3 TIME=11 indicates that the time period of each observation is set by variable 3, and the maximum number of time periods for an individual is 11.

FORMAT = XXX

where XXX is explained below. The default is the integer 0. This option determines the input form of the data. Observations are read either one at a time, with VARI variables read per observation, or as an entire observation matrix.

- FORMAT = 0 With this option observations are read one at a time with a single precision binary read. If this option (cheapest) is used and no print is desired, the program will bypass the observation read section.

- FORMAT = (...) If this option is selected observations will be read one at a time by the format listed after the equal sign. The format should be identical to a FORTRAN statement, and must begin and end with parenthesis, e.g., FORMAT = (7X,4F7.3). If the format continues for more than one line a $ break can be used.

- FORMAT = USER This option implies that observations will be read one at a time by the user-supplied SUBROUTINE USER. USER is a bare-bones subroutine which can be altered to do data transformations, etc.. It is called once for each observation. A listing of SUBROUTINE USER is given elsewhere in the manual.

- FORMAT = USEO This option implies that observations will be read as an entire observation matrix by the user-supplied SUBROUTINE USEO. USEO is a bare-bones subroutine which can be altered to do data transformations, etc.. It is called once. A listing of SUBROUTINE USEO is given elsewhere in the manual.

INPUT = # where # is any integer > 0. INPUT is the input device number (FORTRAN) used to read in observations. It is the number which is assigned to the variable INN in SUBROUTINES USER or USEO. The default is the device used to read in the control cards (generally set to 5).

DISK = # where # is 0, 1, or 2. The default is 0. This option determines internal disk usage.

- If DISK = 0 observations are stored in core (can take a lot of core).

- If DISK = 1 observations are assumed to be stored on a single-precision binary disk file already set up by the user. If this option is used, also use the option FORMAT = 0. The DISK = 1 option can only be used when no DATA TRANSFORMATION(S) are being done (i.e., TRANS must be equal to 0). This option should not be used with a read to end-of-file (OBSV=0). If time periods are set by a variable (e.g. TIME = V3), then the maximum number of time periods per cross-section unit must also be specified (e.g. TIME = 4). If DISK = 1, and PRINT = 0 the program bypasses the observation processing section of the run. This saves a read of the observation file. In this case, the program will use the users input file in the same way as file INT (see DISK = 2) to store the run observations.

- If DISK = 2 the program will write observations on the binary disk file INT (set to 18) and not use core storage. If INT is saved, it can be used for a subsequent run with the option DISK = 1.

LINE = # where # is 0 or 1. The default is 1. This option determines the width of printed output. If LINE = 0 then print width is the normal 132 characters for computer output. If LINE = 1 then print lines are compressed to be 80 characters long. This option is suitable for viewing on a terminal.

PRINT = # where # is 0, 1, 2, 3, 4, 5, 6, or 7. The default is 0. This option determines the print of general variable statistics.

- If PRINT = 0 neither observations nor sample summary statistics are printed.
- If PRINT = 1 only sample variable means and standard deviations are printed.
- If PRINT = 2 sample variable means, standard deviations, and the observation vectors are printed.
- If PRINT = 3 sample variable means, standard deviations, and moment matrix are printed.
- If PRINT = 4 sample variable means, standard deviations, moment matrix and observation vectors are printed.
- If PRINT = 5 sample variable means, standard deviations, and correlation matrix are printed.
- If PRINT = 6 sample variable means, standard deviations, moment matrix and correlation matrix are printed.
- If PRINT = 7 sample variable means, standard deviations, moment matrix, correlation matrix and observation vectors are printed.

MISS < # or MISS = # or MISS > #

where # is any real number not expressed in scientific notation. This option specifies the program's missing value code. The default (if MISS is not specified) is no missing value code. If the value of any input variable violates the missing variable code it will not be included in the mean, moment matrix or correlation computations. It is also possible by use of the MISS code on the MASTER CONTROL CARD to exclude observations with missing values from analysis. A count will also be kept of the number of "good" observations for each variable. Any transformed variables whose computation depends on a variable with a missing value will also be assigned the missing value. The code < and > stand for greater than or equal and less than or equal. Thus MISS<-9999 means that all variables with a value less than or equal to -9999 are assumed to be missing.

NAME = # where # is 0 or 1. The default is 0. This option determines if variables are numbered or given alphanumeric names which are read in.

- If NAME = 0, variable names from 1 to 8 characters apiece are read in on the VARIABLE NAMES CARD(S) for each variable. Names must be read in for all variables input (VARI) on the first VARIABLE NAMES CARD. This card follows the RUN TITLE CARD. If new variables are created on the DATA TRANSFORMATION CARD(S) which are not named there, their names are read on a second VARIABLE NAMES CARD. This card immediately follows the last DATA TRANSFORMATION CARD. If names are used, variables can be referred to by either their numbers or names when specifying equations etc. When referred to by name, the name must be bracketed by [and] in some locations. If referred to by name,

the variable names assigned must be unique or the wrong variable may be used.

- If NAME = 1 variables will be numbered, and the VARIABLE NAMES CARD(S) are not read. Variables must always be referred to by number.

TRANS = # where # is any non-negative integer. The default is 0. TRANS determines the total number of variable transformations read on the DATA TRANSFORMATION CARD(S). If DISK = 1, TRANS must equal 0.

3.3. RUN TITLE CARD

Read a 80 column alphanumeric heading or title which is used on all job print output.

3.4. VARIABLE NAMES CARD PART 1

This card is read only if NAME = 0 on the DATA CONTROL CARD. If used, one to eight character alphanumeric names should be read for each of the VARI variables input for the run. The order of the names should correspond to the order of variables in the input observation vector. If variables are to be referred to by name in specifying equations, their names should be unique and not repeated. Each name should be separated by a blank space or comma (thus commas and blanks cannot be used as part of a name). Include as many lines as necessary to read all variables (do not use the $ continuation, just add extra lines as necessary). Names can use upper or lower case letters. Almost all characters can be used in names except the following: (1), names beginning with V or NV followed by a number (e.g. NV43 or V2); (2), the characters = or - or $ can be used; however, any variables with these characters in them must always be referred to by number or have their name in [and] when used; (3), the names STOP and SAMPLE should not be used.

3.5. DATA TRANSFORMATION CARD(S)

One line or "card" is read for each of TRANS transformations (none if TRANS = 0). This option allows transformations of variables. Each transformation has the following pattern. The DATA TRANSFORMATION CARD begins with an optional FOR statement. It has the following form:

FOR($\#_1$ TO $\#_2$)

where:

$\#_1$ and $\#_2$ are positive integers with $\#_2$ at least as large as $\#_1$.

The FOR statement is very similar to a FORTRAN "DO LOOP". The transformation that follows the FOR statement can contain an index variable designated by the character "I". The transformation will be performed for each value of "I" ranging from $\#_1$ to $\#_2$. "I" can also be used in simple arithmetic expressions with integers, such as I-4 or 6*I-7+I/2. Anytime this

occurs (when "I" does not stand alone), the arithmetic expression should be enclosed by < and >. This may be especially useful in using the "I" index for a variable number. The FOR statement is followed by an optional IF statement which has four possible forms:

IFTIME($\#_1$ TO $\#_2$)

IFCROS($\#_3$ TO $\#_4$)

IF($V\#_5$.XX.$V\#_6$)

IF($V\#_5$.XX.$\#_7$)

where:

$\#_1$-$\#_6$ are integers;

$\#_7$ is any real number not expressed in scientific notation;

XX is one of the logical operators (GT,LT,GE,LE,EQ,NE).

IFTIME specifies a range of time periods and IFCROS specifies a range of cross-sectional units. Defaults for $\#_1$ and $\#_3$ are 1, for $\#_2$ is TIME, and for $\#_4$ is OBSV. $V\#_5$ is any variable read in or computed on an earlier DATA TRANSFORMATION CARD. $V\#_6$ is defined similarly. Variables can be referred to either by their number preceded by a V, or by their name bracketed by [and] (i.e. V12 or [DEPVAR]). The logical operators have conventional FORTRAN treatment, e.g. GT means greater than. The IF statement will be true if a time-period or cross-sectional unit is in the specified range or the logical comparison is satisfied.

Any of the above IF type statements must be followed by a variable specification which determines the number of the new variable:

a * (or /) b ** c = (i.e. ab^c)

where:

a = # (any real number not expressed in scientific notation), *and*

b = V# (the variable to be computed. If the variable has been read in or previously created, it can be referred to by its number preceded by a V or by its name bracketed by [and]. If the variable is new, it can be assigned a number or be given a name here. If the name is assigned here it should be bracketed by [and]. If named here, new variables will be given numbers one larger than the previously largest named variable. This means that any new unnamed variables (referred to here by number) should be given numbers one larger than the sum of all input and named new variables. Names for all new unnamed variables should be read in on a VARIABLE NAMES CARD following the last DATA TRANSFORMATION CARD.), *or*

b = LNV# (natural log of the variable (numbered or named with [and]) to be computed), *or*

b = EXPV# (e raised to the power of the variable (numbered or named with [and]) to be computed).

c = # (any real number not expressed in scientific notation)

Both **a** and c are optional. **b** is not.

If the IF statement is true (or there is no IF statement) the transformed variable will be set by the equation. Otherwise the transformed variable will have its previous value. If the transformed variable is a new variable the default value is 0.

If set by the equation, the value of the transformed variable is determined by the expression following the = sign. This right-hand-side is made up of combinations of "elements" separated by +, −, * or /. Each element has the following general form:

a * (or /) b ** c (or -c) **(i.e. ab^c)**

where:

a = # (any real number not expressed in scientific notation), *and*

b = LN# (natural log of any positive number), *or*

b = EXP# (e raised to the power of any real number not expressed in scientific notation), *or*

b = V# (any variable (numbered or named with [and]) read in or specified on a previous transformation), *or*

b = LNV# (natural log of any variable (numbered or named with [and]) read in or specified on a previous transformation), *or*

b = EXPV# (e raised to the power of any variable (numbered or named with [and]) read in or specified on a previous transformation), *or*

b = V#(+$\#_1$) (any variable (numbered or named with [and]) read in (transformed variables cannot be used) led $\#_1$ time periods. This variable can also be logged or raised to e power. If the time lead extends beyond the maximum number of time periods, the variable will be set to zero. The "+" is optional), *or*

b = V#(−$\#_1$) (any variable (numbered or named with [and]) read in or specified on a previous transformation lagged $\#_1$ time periods. This variable can also be logged or raised to e power. If the time lag extends beyond the first time period, the variable will be set to zero), *or*

b = VT0 (a time trend with value equal to the time period. This can be logged or raised to e power), *or*

b = VT# (a dummy variable which has a value of zero unless the time period equals # when it is set to one. If VT# is zero then the element will be set to zero),

or

b = (...) where the ... are other elements with a single layer of parenthesis, *or*

b = LN(...) natural log of what is in the parenthesis, *or*

b = EXP(...) e raised to the power of what is in the parenthesis, *and*

c = # (any real number not expressed in scientific notation. There is a restriction however. If "c" is larger in absolute value than 1, it will be rounded to the nearest integer), *or*

c = V# (any variable (numbered or named with [and]) either read in or specified on a previous transformation), *or*

c = V#(+$\#_1$) (any variable (numbered or named with [and]) read in (transformed variables cannot be used) led $\#_1$ time periods. If the time lead extends beyond the maximum number of time periods, the element will be set to zero. The "+" is optional), *or*

c = V#(-$\#_1$) (any variable (numbered or named with [and]) read in or specified on a previous transformation lagged $\#_1$ time periods. If the time lag extends beyond the first time period, the element will be set to zero)

Any of the terms **a, b, c** are optional.

Some examples are:

```
IF(V22.Gt.6)V29 = 4*V28(-1)**2
V30 = EXPVT0*V31(+1)/1000
IFCROS(1TO50)V34 = LN(V8 + V6)*EXP(V11 + V12)**-V4
-2.73*V39**.5=V1**2+V2**2/17.
```

The following two lines mean the same thing,

```
IFTIME(6TO6)V33 = V9*V10/V11 + LNV12
V33 = (V9*V10/V11 + LNV12)*VT6
```

The following two lines can form two IF statements having the effect of an .AND. operator,

```
IF(V1.GT.7)V35 = VTO + 17
IFTIME(6TO8)V35 = 0
FOR(2TO4) V<I+2>=I
```

The following two lines also mean the same thing,

```
FOR(2TO4) IFTIME(ITOI) V<35+I>=1.
FOR(2TO4) V<35+I>=VTI
FOR(11TO20) VI=V<I-11>(-1)
FOR(1TO5) V<5+I>=V1(-<I+1>)
```

If V2 is named DEPTOB and V3 is named REEDER these are the same:

```
IF(V2.GT.3) 3*V39**2=V3*V11-LNV12
IF([DEPTOB].GT.3) 3*V39**2=[REEDER]*V11-LNV12
```

The following is the same as above but names the new V39 here Its name should not then be read on the VARIABLE NAMES CARD,

```
IF([DEPTOB].GT.3) 3*[NEW]**2=[REEDER]*V11-LNV12
```

3.6. VARIABLE NAMES CARD PART 2

This card is read only if NAME = 0 on the DATA CONTROL CARD, TRAN > 0, and new unnamed variables are created on the DATA TRANSFORMATION CARD(S). Names are read for any such variables, in order, using exactly the same rules as specified for the VARIABLE NAMES CARD PART 1. If there are no such variables, this card is not read.

3.7. MASTER CONTROL CARD

The following options are specified on the MASTER CONTROL CARD. They can be listed in any order and can be omitted if the user wishes the option to take on its default value. Options may be continued onto several lines so long as a $ is used at the end of a continuation line.

We first consider the "essential" commands; namely those commands which specify the number of equations, model type, and basic estimation form. We then consider options which govern parameter restrictions, equation non-linearities, and likelihood weighting. The next set

of commands controls the specification of the structure of equation error covariances and the computation of standard errors. These are followed by options used with instrumental variable estimation and then testing. Commands which govern which observations will be selected for the run are specified in the next set of options. Then options which determine the starting values, scaling, tolerances, and convergence criteria for iterative solution methods are listed. Finally, commands are given that control what print and punch output will be produced by the run.

3.7.1. The STOP, EQUA, MODEL and TYPE Options

The following commands indicate the number of equations to be estimated jointly, what type of model specification is to be estimated and the estimation technique to be employed. We now describe each command in detail indicating the possible options for each command.

Option	Description
STOP	This command instructs the program to stop. It should be placed at the end of the control file. The entire command STOP must be spelled out (no abbreviations).
EQUA = #	where # is any integer > 0. The default is 1. EQUA is the total number of estimated equations in the system.
MODEL = XX	where XX is explained below. This option determines the model type if the same model applies to all equations. The default is to determine model type individually equation by equation on the BASIC EQUATION CARD. The models allowed are (see Section 2.1 for a more complete description):

- MODEL = PROBIT. The model(s) used assumes a 0/1 probit dependent variable.
- MODEL = TOBIT. The model(s) used assumes a Tobit dependent variable.
- MODEL = REGRES. The model(s) used assumes a continuous observed dependent variable (regression).
- MODEL = IMPLICIT. The model(s) used assumes an implicit function with no dependent variable. This is equivalent to MODEL = REGRES with a dependent variable always equal to zero. It should not be used without parameter restrictions.
- MODEL = DBLTOB. The model(s) used assumes a Tobit dependent variable with an upper and lower mass point.
- MODEL = TRUNCATE. The model(s) used assumes a truncated normal dependent variable. This can be used only if TYPE = 1 or 3.
- MODEL = LOGIT. The model(s) used assumes a 0/1 logistic dependent variable. This can be used only if TYPE = 1 or 3.

TYPE = # where # is 1, 2, 3 or 4. There is no default. This option determines the type of estimation procedure used (see Section 2.2 for a more complete description).

- If TYPE = 1 then the program uses the cheapest option, "simple" maximum likelihood (ML). Except for cross-equation restrictions, each equation is estimated separately, and even with panel data no account is taken of potential correlation of errors over time (except for standard errors which can be adjusted). Instrumental variables cannot be used. This is equivalent to single-time period single-equation i.i.d. ML. If equation and/or inter-temporal errors are correlated, parameter estimates will generally still be consistent, but they will only be "quasi ML".

- If TYPE = 2 then the program uses single factor loading ML estimation utilizing the Butler-Moffitt procedure, or a first order autoregressive procedure. There are three models that can be estimated:

  - If there are multiple time periods then the program can estimate an equi-correlated error structure over time, a so-called "random effects" model. The parameter estimated will be the error correlation across time periods. If there are multiple equations a separate parameter is estimated for each equation. For this selection set the variable RANDOM = 0.

  - If there are multiple time periods the program can also estimate a first order autoregressive error model. The parameter estimated will be the error correlation of two observations one period apart. The correlation of errors two periods apart is the parameter squared etc. If there are multiple equations a separate parameter is estimated for each equation. For this selection set the variable RANDOM = 1.

  - If there are multiple equations, then this option can be used to estimate cross-equation error correlations as loadings on a single normal error scaled to have a variance of one. The parameters estimated in this case will be each correlation between the factor error and the equation error. The correlation between two equation errors can then be computed as the product of their factor loadings. For this selection set the variable RANDOM = 2. Note that this model can also be estimated by using TYPE = 4 and imposing restrictions on the error covariance structure. Particularly for regressions and Tobits, the TYPE = 4 method can be considerably cheaper.

  Warning - It generally is necessary to impose a restriction to normalize factors if RANDOM = 2 (see RESTRI). The nature of the normalization can effect results. If there are both multiple time periods and equations any of the three options can be selected. TYPE = 2 can't be used with TRUNCATE or LOGIT.

- If TYPE = 3 then the program allows for the use of instrumental variables (IV). The program further allows for the potential optimal

weighting of the orthogonality conditions arising from the use of instruments (see OPTWT). If there is only one time period, this option generally reduces to single-equation single-time period i.i.d. ML (TYPE =1). There are two exceptions, however. With non-linear equations (NONLIN > 0), although the equation residuals will be computed using the non-linear form, the instruments used will be the specified X's. This will differ from TYPE = 1 which implicitly uses gradients of the non-linear function as instruments. Similarly, if instruments are used which are not explicitly in the equation, TYPE = 3 will differ from TYPE = 1. These could include other variables, or values of equation variables in other time periods.

- If TYPE = 4 then the program estimates multiple-equation systems taking into account correlations of their errors via bivariate ML. Intertemporal errors with panel data will be ignored except for standard errors. Equations will be estimated using a quasi-ML procedure formed from bivariate likelihood functions formed from all pairs of equations. This option should not be used with one equation. With two equations it is equivalent to ML. Equation error correlations (covariances) will be estimated as well as coefficients. TYPE = 4 can't be used with TRUNCATE or LOGIT.

3.7.2. The NONLIN, RESTRI, CONST and WEIGHT Options

The following four options are used to specify the nature of the estimated equations and/or parameter restrictions. They will to a large extent determine the cost and complexity of a run. The NONLIN option determines whether the system equations (and error parameters) are linear or non-linear in the x's. The RESTRI command determines whether or not there are restrictions in the parameters. The WEIGHT and CONST commands determine whether observations are weighted and if a constant is used in equations.

Option	Description

NONLIN = # where # is 0, 1, or 2. The default is 0.

- If NONLIN = 0, models are linear in the X's (except for Tobit, probit transforms) and are specified on the EQUATION VARIABLE CARD. This is the cheapest option, and should be used if system non-linearities are restricted to non-linearities in the parameters (see RESTRI). The program will use moment-based calculations for REGRES models and the non-mass point observations for TOBIT and DBLTOB.

- If NONLIN = 1 then equations, error variances, and intercorrelations are assumed to be non-linear in the X's and are specified on the NON-LINEAR EQUATION CARD(S). If NONLIN = 1 then starting values must be read in (START = 0, 1). No starting values should be zero, however, as this will sometimes cause gradient problems. This option is the most expensive and should be used only when necessary. It must be used, however, when any equation error variance or covariance is a function of the X's. It should not be

necessary under any other circumstances (use NONLIN = 2).

- If NONLIN = 2 then only equations are non-linear in the X's and specified on the NON-LINEAR EQUATION CARD(S). Equation error variances, and intercorrelations are computed linearly; thus, estimation is cheaper. These terms will be assigned parameter numbers following those specified in the non-linear equations. This option is almost identical to NONLIN = 1 for starting values (START = 0, 1, 3). It should be used if any system equation is non-linear in the X's but no error parameters are.

RESTRI = # where # is any non-negative integer. The default is 0. This option determines how many restrictions are imposed on parameters in estimation. Each restriction is specified separately on the RESTRICTION CARD(S). Any subset of the estimated parameters (including variances, etc.) can be restricted to be a function of the unrestricted parameters, with estimates and standard errors appropriately adjusted. This option can be used with both linear and non-linear equations. If all system non-linearities can be expressed as non-linearities in the parameters, it is strongly advised that RESTRI, not NONLIN, be used. This allows the system to use the cheaper computation methods of NONLIN = 0, particularly when moment-based estimators can be employed. If restrictions are used, restricted coefficients will be marked with an "R" on printed output.

CONST = # where # is 0 or 1. The default is 0. This option determines if a constant is used in each model equation. If CONST = 0, it is. If CONST =1 it is not. This option can be overridden on an equation by equation basis by adding a constant as variable V0 (or CONSTANT) or taking one out with NV0 (or N[CONSTANT]). For fixed effect models (FIXED = 1 on the BASIC EQUATION CARD) the default will be that the constant is not included. The CONST option only applies to equations or orthogonality conditions specified on the EQUATION VARIABLE CARD. It does not apply to the NON-LINEAR EQUATION CARD(S) where the constant must be specified manually (or included as a parameter).

WEIGHT = V#
where # is any integer from 1 to VAR*. This option determines if the sample log-likelihood is weighted. A variable name bracketed by [and] can also be used. If WEIGHT is not specified there is no weighting. The value in variable V# is used to weight each equation and time period likelihood. It should be constant over time, therefore, if TYPE = 2. Only positive weighting values are allowed. Thus all observations with zero or negative values for the weighting variable will be dropped. The weights will be scaled to average one so as not to effect the log-likelihood scale.

3.7.3. Options for Disturbance Term Specification

The following options allow the user to choose several features of the the stochastic disturbance structure to be estimated for a particular model. For example, the option RANDOM allows one to choose whether one wants a factor analytic or autoregressive covariance structure for equation disturbances, while options TCORR and ECORR control whether or not to com-

pute intertemporal and across equation covariances. The options SIGMA and RHO determine whether or not variances or standard deviations of disturbances and covariances or correlations of disturbances are estimated. The options PSTAN and MAVER determine how parameter standard errors are computed. They offer a fair amount of flexibility with multiple time-periods or equations.

Option	Description

RANDOM = #where # is 0, 1, or 2. The default is 0. This option applies only if TYPE = 2, and determines the type of factor loading or autoregressive model estimated. See TYPE = 2 for an explanation.

HERMIT = # where # is 1-4, 6, or 9. The default is 6. This option applies only if TYPE = 2 and RANDOM = 0 or 2. It determines the number of Hermite points used in the approximate numerical integration for single factor loading ML. The more Hermite points used, the more accurate is the approximation, but the more expensive is the computation. Nine points are the maximum number of points that can be specified. [See Abramowitz and Stegun (1972, p. 924).]

TCORR = # where # is 0, 1, 2, or 3. The default is 0. This option determines if correlations of model errors are computed across time periods. These estimates are computed completely separate from the coefficients using a conditional "quasi-ML" likelihood function. It does not apply if TIME = 1 or TYPE = 2,4. If there are multiple equations, separate correlation matrices are computed for each equation. This option can be used if coefficients are input or estimated as part of the same run. It is generally best run separately, however.

- If TCORR = 0, the program does not compute inter-temporal error correlations.

- If TCORR = 1, the program computes the inter-temporal correlation of each equation's errors using quasi-maximum likelihood. (It maximizes only over the correlations and takes coefficients as input or computed in a previous step). Starting values for the correlations are read on the TIME PERIOD CORRELATION CARD. Estimation of the coefficient vector is determined by TYPE and ESTIMA. If ESTIMA < 4 then coefficients will be estimated as part of the same run, otherwise they must be input.

- If TCORR = 2, same as option (=1) except that starting values for the time period correlations are computed from the observed model errors after coefficient estimates are computed (or input).

- If TCORR = 3, same as options (=1) or (=2) except that the coefficient vector is read in and is not estimated. Variable ESTIMA applies to the time-period correlation estimation. Coefficient and correlation values are read by a fashion determined by the command START to be described below. If START = 0, 2 coefficients are read on the COEFFICIENT STARTING VALUE CARD, and are also used for the weighting matrix. If START = 1, 3 coefficients are read as well as a separate coefficient vector for the weighting matrix on the WEIGHTING MATRIX COEFFICIENT CARD. If START = 0, 1 time period correlation starting values are read on the TIME

PERIOD CORRELATION CARD. If START = 2, 3 time period correlation starting values are computed from observed model errors. If TCORR = 3 and ECORR = 3, START must apply to both.

ECORR = # where # is 0, 1, 2, or 3. The default is 0. This option determines if correlations of model errors are computed across equations. These estimates are computed completely separate from the coefficients using a conditional "quasi-ML" likelihood function. It does not apply if EQUA = 1 or TYPE = 2, 4. Only one matrix of equation correlations is computed for all time periods. This option can be used if coefficients are input or estimated as part of the same run. It is generally best run separately, however. The options are identical to those of TCORR.

- If ECORR = 0, the program does not compute inter-equation error correlations.
- If ECORR = 1, the program computes the correlation of errors across equations using quasi-maximum likelihood. (It maximizes only over the correlations and takes coefficient vectors as input or computed in a previous step). Starting values are read on EQUATION CORRELATION CARD. Estimation of the coefficient vector is determined by TYPE and ESTIMA. If ESTIMA < 4 then coefficients will be estimated as part of the same run, otherwise they must be input.
- If ECORR = 2, same as option (=1) except that starting values of the inter-equation correlations are computed from observed model errors after coefficients are computed (or input).
- If ECORR = 3, same as options (=1) or (=2) except that the coefficient vector is read in and is not estimated. The command ESTIMA applies to the inter-equation correlation estimation. Coefficient and correlation values are read by a fashion determined by the variable START. IF START = 0, 2 coefficients are read on the COEFFICIENT STARTING VALUE CARD, and are also used for the weighting matrix. If START = 1, 3 coefficients are read as well as a separate coefficient vector for the weighting matrix on the WEIGHTING MATRIX COEFFICIENT CARD. If START = 0, 1 inter-equation correlation starting values are read on the EQUATION CORRELATION CARD. If START = 2, 3 inter-equation correlation starting values are computed from observed model errors. If ECORR = 3 and TCORR = 3, START must apply to both.

SIGMA = # where # is 0, 1, 2, 3, or 4. The default is 0. This option determines the general treatment of model error variance terms. It can be overridden on an equation by equation basis on the BASIC EQUATION CARD or the NONLINEAR EQUATION CARD(S).

- If SIGMA = 0 then model error variances are scaled to 1 and not estimated for LOGIT and PROBIT. The standard deviation of model errors is estimated for TOBIT, DBLTOB, TRUNCATE, IMPLICIT, or REGRES.

- If SIGMA = 1 then the standard deviation of the model errors is estimated for PROBIT (generally requires parameter restrictions). This option does not apply for LOGIT. Same as option (=0) for other models.

- If SIGMA = 2 then the same as option (=1) except that model error variances are estimated for each model (including PROBIT).

- If SIGMA = 3 then the same as option (=0) for all models except when TYPE = 3 and regression, implicit, truncated or Tobit equations are estimated. In these instances, the equation error sigma is specified however, the sigma orthogonality condition is not imposed. Thus sigma applies only for scale. If used, sigma generally must be restricted (see RESTRI).

- If SIGMA = 4 then the same as option (=3) except that model error variances are specified instead of error standard deviations (including PROBIT).

RHO = # where # is 0 or 1. The default is 0. This option determines if inter-equation error correlations or covariances are estimated when TYPE = 4. It applies only if TYPE = 4 and can be overridden on the NON-LINEAR EQUATION CARD(S).

- If RHO = 0 then the program estimates inter-equation error correlations.

- If RHO = 1 then the program estimates inter-equation error covariances.

PSTAN = # where # is 0, 1, 2, or 3. The default is 0. This option determines the method used to calculate standard errors of coefficients. Options (=1) (=2) or (=3) should be relevant only if multiple time periods or equations are used.

- If PSTAN = 0 and TYPE = 1, 2, 4 then the program computes standard errors as if observations were I.I.D. (generally incorrect with panel data). If TYPE = 3 and OPTWT = 0 this means that the program computes standard errors under the assumption of optimal weighting (see equation (4.1.18)). For TYPE = 1 or 4 the program will use the inverse matrix of the negative of the sample log-likelihood 2nd derivative matrix evaluated at the estimated parameters. If TYPE = 2 the program uses the inverse of the cross-products matrix of the log-likelihood 1st derivatives (for RANDOM = 0 or 1 the first derivatives are first summed over all time periods for each cross-section before they are cross-producted).

- If PSTAN = 1 and TYPE = 1, 3, 4 then the program corrects standard errors for inter-temporal correlations (time-adjusted). If non-panel data is used this should have the same properties as option (=0). The parameter covariance matrix is given by D^{-1} S D^{-1} where D is the negative of the matrix of log-likelihood 2nd derivatives and S is a cross products matrix of 1st derivatives. This option can also

be used if TYPE = 2, but the D matrix used will be the same as that used with option (=0). With options RANDOM = 0 or 1, or with non-panel data, or with option MAVER = 0, option (=1) will give equivalent estimates to option (=0) for TYPE = 2. With TYPE = 3 the formula is very similar except orthogonality conditions are used instead of 1st derivatives and gradients of the orthogonality conditions are used instead of 2nd derivatives (see equation (4.1.19)). Generally the 1st derivatives (or orthogonality conditions if TYPE = 3) will be summed over all time periods for each cross-section before they are cross-producted. This option can be changed, however, if time-independence or a moving average time correlation specification is desired, by the option MAVER.

- If PSTAN = 2 then the program computes standard errors both ways by options (=0) and (=1).

- If PSTAN = 3 and TYPE = 1 or 4 then the program computes standard errors from the inverse of the summed cross-products matrix of sample log-likelihood 1st derivatives evaluated for each cross-section (matrix S above). If estimates are maximum likelihood these standard errors will have equivalent properties to those given by option (=0) or (=1). Generally with panel data the derivatives will be summed over time periods before they are cross-producted, but this option can be changed with option MAVER. One disadvantage with this option is that the program will not print "measure of fit" statistics.

MAVER = # where # is any integer from 0 to TIME - 1. With panel data the default is TIME - 1. With pure time series or cross-sectional data the default is 0. This option determines the number of time period leads and lags used in summing log-likelihood first derivatives (or orthogonality conditions if TYPE = 3) when the cross-products matrix of 1st derivatives is used in standard error calculations (or TYPE = 3). This option applies only if TIME > 1 and PSTAN = 1 or 3 (or PSTAN = 0 and TYPE = 3). If MAVER = TIME - 1 then the program will sum derivatives for all time periods of each cross-section before cross-producting. If MAVER = 0 then derivatives will be cross-producted for each time period (generally correct only if equation errors are independent over time). If MAVER is between 0 and TIME - 1 then derivatives will be summed for MAVER time periods preceding and lagging each time period observation. This option is correct under the assumption that equation errors are correlated with a moving average process of order MAVER.

3.7.4. Options for IV Estimation

The following options control the use of weighted IV estimation and the number of leads and/or lags used if instruments from other time periods are used. It only applies if TYPE = 3.

Option	Description
OPTWT = #	where # is 0, 1, or 2. The default is 0. This option applies only if TYPE = 3, and determines which weighting scheme is used.

- If OPTWT = 0 the program uses input coefficient values to compute an estimate of the optimal weighting matrix of the orthogonality conditions arising from the instruments which is used unchanged for all estimation. If there are overidentifying orthogonality conditions, a chi-square test of the model will automatically be performed.

- If OPTWT = 1, orthogonality conditions are weighted equally with an identity matrix, and not optimally. This option can be used like OPTWT = 0 when there are auxiliary instruments or orthogonality conditions. Note, however, that unlike OPTWT = 0 it can be sensitive to scale.

- If OPTWT = 2, then a preliminary regression will be run for all included independent variables in each equation against the list of instruments for the equation. Coefficients from these regressions will be used to weight the orthogonality conditions. This option is equivalent to traditional instrumental variables or two-stage least squares. It should be used only if there are endogenous variables in the equation which are not used as instruments. It cannot be used if NONLIN is not 0. (This option is not yet operational)

LAGS = # where # is any integer from 0 to TIME - 1. The default is TIME - 1. This option applies only if TYPE = 3, and determines the number of time leads and/or lags used for instruments for those variables allowed to have non-contemporaneous orthogonality conditions. If LAGS = 0 the program defaults to LAGS = TIME - 1. Although not necessary from the program's standpoint, it will generally make sense for the number of lags used in computing standard errors (MAVER) to be at least as large as LAGS.

3.7.5. Options for Testing With IV Estimation

The following options control what sorts of test statistics will be computed for each equation or system of equations used with IV estimation (TYPE = 3).

Option	Description

TEXOG = # where # is 0 or 1. The default is 0. This option applies only if TYPE = 3, and determines whether or not there are exogeneity tests. If TEXOG = 1, then the program will perform exogeneity tests using the estimated coefficients. Orthogonality conditions (instruments) read on the EQUATION TESTING VARIABLE CARD are tested against these used to estimate coefficients on the EQUATION VARIABLE CARD. Orthogonality conditions specified on the latter must always be nested in the former. Different lead/lag structures and/or cross-equation orthogonality conditions can be tested. Even if no explicit tests are desired, TEXOG can be used if the model is over-identified. The program will test the over-identifying restrictions and print out the value of each orthogonality condition. If this is desired, the EQUATION TESTING VARIABLE must still be included, but should be left blank. Orthogonality conditions will be identified on printed output as contemporaneous (0), leads (+), and lags (-). Variables will be listed by their numbers.

TSUB = # where # is any non-negative integer. The default is 0. This option applies only if TEXOG = 1. It specifies the number of different Wald-type chi-square tests which are performed on subsets of the orthogonality conditions specified on the EQUATION TESTING VARIABLE CARD. If TSUB > 0 then a separate ORTHOGONALITY CONDITION TEST CARD is read for each of the TSUB tests.

TLAG = # where # is any integer from 0 to TIME - 1. The default is LAGS. This option applies only if TEXOG = 1, and determines the number of leads and/or lags of non-contemporaneous orthogonality conditions used in exogeneity tests specified on the EQUATION TESTING VARIABLE CARD. TLAG must be greater than or equal to LAGS.

3.7.6. Options for Missing Data and the Inclusion of Observations in Estimation

The following options control how observations with missing data will be handled and what observations and/or time periods (relevant for panel data sets) will be included in the estimation of an equation (or system of equations) and how they will be stored.

Option	Description

MISS = # where # is 0, 1, or 2. The default is 0. This option determines the treatment of observations with missing values. It does not apply unless MISS is also specified on the DATA CONTROL CARD.

- If MISS = 0 (here), then no observations will be excluded because of missing data.
- If MISS = 1 (here) then if an equation is missing any independent or dependent variables for an observation, the observation will not be used for that equation.
- If MISS = 2 (here) then if any equation has a missing independent or dependent variable, the observation will not be used for any equations. Both MISS = 1 and MISS = 2 apply on a time period by time period basis with panel data. Only MISS = 2 can be used if NONLIN is not 0.

DISK = # where # is 0 or 1. The default is 0. This option determines internal disk usage for this model run only. It specifies a different option than DISK on the DATA CONTROL CARD.

- If DISK = 0 model run data is stored in core. This takes more core storage but is generally much cheaper in terms of CPU.
- If DISK = 1 the program will write model run data on the binary disk file INZ (set to 19) and not use as much core storage. This option should not be used with a pure time series as disk storage will save no core (each disk write is equivalent to one cross-sectional unit, hence the entire observation file for a time series). If START

= 2 or 4 and disk storage is used, then observations cannot be scaled (SCALE must = 1). When users have virtual machines it might appear that full core storage is always desirable. However, particularly with panel data, the programs own swapping mechanisms with DISK = 1 are often cheaper than the system swaps.

UTIME = $\#_1$ TO $\#_2$

where $\#_1$ is any integer from 1 to TIME and $\#_2$ is any integer from $\#_1$ to TIME. The default for $\#_1$ is 1 and for $\#_2$ is TIME (or the maximum number of time periods if variable time is used). This option determines the range of time periods used for the run. It applies to all equations.

XTIME = $\#_1$ TO $\#_2$

where $\#_1$ is any integer from 1 to TIME and $\#_2$ is any integer from $\#_1$ to TIME. This option determines the range of time periods which will not be used in the run for any equations. It cannot be used with UTIME.

UCROS = $\#_1$ TO $\#_2$

where $\#_1$ is any integer from 1 TO OBSV and $\#_2$ is any integer from $\#_1$ to OBSV. The default for $\#_1$ is 1 and for $\#_2$ is OBSV. This option determines the range of cross-sectional observations used for the run. Any units not in the specified range will not be used for any equations. With non-panel data it determines which observations are used.

XCROS = $\#_1$ TO $\#_2$

where $\#_1$ is any integer from 1 TO OBSV and $\#_2$ is any integer from $\#_1$ to OBSV. This option determines the range of cross-sectional observations which will not be used for the run for any equations. It cannot be used with UCROS.

UOBSV = V$\#_1$.XX.V$\#_2$ *or* UOBSV = V$\#_1$.XX.$\#_3$

where $\#_1$ and $\#_2$ are any integers from 1 to VAR*, $\#_3$ is any real number not expressed in scientific notation, and XX is one of the logical operators GT, LT, GE, LE, EQ, NE. There are no defaults. Variable names bracketed by [and] can be substituted for V$\#_1$ and/or V$\#_2$. If UOBSV is not specified, no observations will be excluded on this basis. This option specifies a FORTRAN IF statement which determines which observations are used in a run. Any observations not meeting the criteria will not be used for any equations. V$\#_1$ specifies a variable whose value is compared either to the value in variable V$\#_2$ or the real number, $\#_3$ using the comparison indicated by the logical operator (conventional FORTRAN treatment, e.g., GT means greater than). If the comparison is true (i.e., V22.GE.3.27 or V27.EQ.V28 or [REEDER].NE.[NEW]) then the observation is used. The comparison is made separately for each time period of each cross-section unit.

3.7.7. Options for Starting Values and Choice of Numerical Optimization Methods

The following options control how starting values of parameters will be chosen and what sort of optimization techniques will be employed. There are also some options to control the

tolerances which govern convergence of parameters and whether or not parameters should be scaled in estimation.

Option	Description

START = # where # is 0, 1, 2, 3, or 4. The default is 0. This option determines the method of calculating coefficient starting values (except when ECORR = 3 or TCORR = 3, see those options).

- If START = 0 starting values are read in on the COEFFICIENT STARTING VALUE CARD. This must be used (or option (=1,3)) if NONLIN is not 0 or MODEL = IMPLICIT. If TYPE = 3, starting coefficients are also used to compute the weighting matrix if OPTWT = 0.

- If START = 1 coefficient starting values and weighting matrix coefficients are read separately. This applies only if TYPE = 3 and OPTWT = 0. The weighting matrix coefficients are read second on the WEIGHTING MATRIX COEFFICIENT CARD.

- If START = 2 the program uses regression methods to get starting values for the models in question. This option cannot be used if NONLIN = 1 or 2. Depending on the type of model specification being estimated these starting values are computed in several different ways.

  - If TYPE = 1 or TYPE = 4 and no fixed effects are used, then starting values are determined by regressing the observed dependent variable (0/1 etc.) against the independent variables for starting values. For logits and probits, the coefficients will then be scaled to reflect differences in their respective scalings and that of the linear probability model.

  - If a fixed effect model is used (FIXED = 1) and TYPE = 1 starting values are computed from a regression fixed effect model (a covariance model).

  - If a random effect model is used (TYPE = 2 and RANDOM = 0) starting values are computed from a two-pass error components regression.

  - If a first-order autoregressive error model is used (TYPE = 2 and RANDOM = 1) starting values are computed from a two-pass Cochran-Orcutt procedure.

  - If Instrumental Variable (IV) estimation is used (TYPE = 3) and OPTWT = 2 starting values are computed from instrumental variable regressions utilizing the observed dependent variable. These are equivalent to two-stage least squares estimates with at least some extra instruments.

  - If coefficient restrictions are imposed (RESTRI > 0) and TYPE = 1 or TYPE = 3, restricted regressions will be run. These

regressions, estimated in a second pass after unrestricted regressions, impose a "linearization" of the restrictions. They will be the correct regressions if indeed the restrictions are linear.

- Correlations of residuals from these equations are used as starting values for inter-equation or time period correlations if TYPE = 4 or TYPE = 2 and RANDOM = 2.

In the case of regression models, these starting values will be consistent, although they will generally not be ML when TYPE = 2 or 4. Although the standard errors may be off, they can be printed in more detail by setting ESTIMA = 4. Care should be used with regression starting values if TYPE = 3 and OPTWT = 0, as the weighting matrix may not truly be optimal as the regression starting values may not be consistent (with PROBIT, TOBIT etc.). The final estimated coefficients will generally still be consistent.

- If START = 3 then the coefficients are read as in option (=0). However these coefficients are used as starting values for TYPE = 1 estimates. The computed TYPE = 1 estimates are then used as starting values (and for the weighting matrix) for TYPE = 2, 3, 4 estimates as specified by TYPE. If TYPE = 2 or 4 the COEFFICIENT STARTING VALUE CARD must also include starting values for inter-equation or time period correlations (this option can't be used if these are needed for coefficient identification or NONLIN = 1).

- If START = 4 the treatment is essentially the same as option (=3). However, the starting values for TYPE = 1 first round estimates are computed by the same procedure as option (=2). Inter-equation or time period correlation starting values are computed from equation residuals. This option cannot be used if NONLIN = 1 or 2.

FORMAT = # where # is 0 or 1. The default is 0. This option determines the format for all starting values (see START, ECORR, or TCORR).

- If FORMAT = 0 then all coefficients are read with (5D16.9) format (same as they are punched). No FORMAT CARDS are read.

- If FORMAT = 1 then the format for each coefficient vector is read preceding each STARTING VALUES CARD on the FORMAT CARD.

SCALE = # where # is 0 or 1. The default is 0. This option determines coefficient scaling.

- If SCALE = 0 then the coefficients are scaled to 1 by starting values. This is done for computational accuracy and does not effect the printout. Variances and correlations are not scaled. Dependent variables should be scaled by hand, therefore, so that error variances are around one as well.

- If SCALE = 1 variables are not scaled. This option must be used if DISK = 1 on the MASTER CONTROL CARD and START = 2 or 4.

OUTLY = # where # is 0, 1, or 2. The default is 0. This option determines how the program handles outliers encountered during estimation.

- If OUTLY = 0 then the program will stop if it encounters an extreme observation in evaluating a probit/logit/tobit/truncated datapoint. This option can be used to avoid bad starting values.

- If OUTLY = 1 the program will set extreme outlier observations to boundary values if encountered, and will continue estimation. The program will also adjust negative variances or correlations out of bounds. Any corrections will be printed.

- If OUTLY = 2 the program will make the same corrections as OUTLY = 1, but will not print the correction.

ESTIMA = # where # is 0, 1, 2, 3, 4, or 5. The default is 0. This option determines the estimation iteration control. See the Appendix on Fletcher-Powell for more details.

- If ESTIMA = 0 the program uses both steepest descent and Fletcher-Powell iteration methods. The program will use the cross-products of the gradients after steepest descent as starting values for the Fletcher-Powell second derivatives matrix. If this is not invertible a partial identity matrix will be substituted.

- If ESTIMA = 1 the program stops after the steepest descent section.

- If ESTIMA = 2 same as option (=0) except that the program only uses Fletcher-Powell, no steepest descent. This option or (=3) should be used if time limits were reached on previous runs and a problem is being restarted in the middle.

- If ESTIMA = 3 same as option (=2) except that an identity matrix is used for starting values of the Fletcher-Powell second derivatives matrix not the cross-products of the gradients.

- If ESTIMA = 4 there will be no estimation at all. The program will compute coefficient standard errors using input values, and will perform exogeneity testing, and residual analysis. This option applies to both coefficient estimation and inter-equation or time period error correlation estimation.

- If ESTIMA = 5 same as option (=4) except that coefficient standard errors are also not computed. Unlike option (=4), this option does not apply to inter-equation or time period error correlations, only to coefficients. If ECORR > 0 or TCORR > 0 the program will estimate correlations using both steepest descent and Fletcher-Powell. Note that if ECORR = 3 or TCORR = 3 then ESTIMA applies only to correlation estimation.

CNVRG = # where # is 0, 1, 2, 3, or 4. The default is 0. This option determines the criteria used for parameter convergence. See Appendix on Fletcher-Powell for

more details.

- If CNVRG = 0 the program requires three criteria to all be satisfied for convergence. These are:

  1. The maximum element of the parameter gradient vector is less in absolute value than GTOL;

  2. The absolute proportional change in the function value is less than FTOL for two iterations (this criteria should not be used with TYPE = 3 with just-identified models); and

  3. The maximum absolute proportion change of any parameters is less than PTOL for two iterations.

- If CNVRG = 1, the program will stop when any of the convergence criteria specified in option (=0) are satisfied.

- If CNVRG = 2, the program will stop only when the gradient convergence criteria is satisfied.

- If CNVRG = 3, the program will stop only when the function change convergence criteria is satisfied.

- If CNVRG = 4, the program will stop only when the parameter change convergence criteria is satisfied.

GTOL = # where # is any real number not expressed in scientific notation > 0. The default is .00001. This option determines the tolerance used for convergence by the maximal absolute element of the parameter gradient vector.

FTOL = # where # is any real number not expressed in scientific notation > 0. The default is .0000001. This option determines the tolerance used for convergence by the absolute proportional change in the function value. It must be satisfied for two iterations.

PTOL = # where # is any real number not expressed in scientific notation > 0. The default is .0001. This option determines the tolerance used for convergence by the maximal absolute proportional change of any element of the parameter vector. It must be satisfied for two iterations.

FETOL = # where # is any real number not expressed in scientific notation > 0. The default is .00001. This option determines the tolerance used for convergence in estimating the fixed effect if FIXED = 1. It does not apply otherwise. The tolerance should be weakened if the program has trouble getting started when estimating a fixed effect model. When trouble is encountered the program will print an error message if OUTLY = 1 or 2.

ITERA = # where # is any non-negative integer. The default is 50. This option determines the total combined number of iterations allowed in the steepest descent and the Fletcher-Powell section of program estimation.

SECS = # where # is any non-negative real number. The default is 5. This option determines the total number of CPU seconds allowed for both steepest descent and Fletcher-Powell sections of the program. This option applies only if a CPU timer is used in the program.

3.7.8. Options Controlling Output Printed and Output Punched

The following options control what information from estimation is printed out to the output and punch files.

Option	Description

PCOVA = # where # is 0 or 1. The default is 0. This option determines if the full covariance matrix of estimated parameters is printed.

- If PCOVA = 0 the matrix is not printed.
- If PCOVA = 1 it is.

IPRINT = # where # is 0 or 1. The default is 0. This option determines if iteration information is printed. If IPRINT = 0, the program prints information on each iteration of the estimation process. If IPRINT = 1, then the program prints only summary information.

CWRITE = # where # is 0, 1, or 2. The default is 0. This option determines whether estimated parameters are punched (written on disk) on device IPC (generally set to 7).

- If CWRITE = 0 no coefficients are punched/written.
- If CWRITE = 1 the program punches/writes final coefficient (and inter-equation or time period correlations if ECORR > 0 or TCORR > 0) estimates on device IPC using format (5D16.9). Coefficient order is the same as printed.
- If CWRITE = 2 same as option (=1) except that coefficients (correlations) are also punched/written at each steepest descent/Fletcher-Powell iteration. This option is useful if the computer run can fail in the middle.

RPRINT = # where # is 0 or 1. The default is 0. This option determines whether the program performs and prints a residual/predicted value analysis.

- If RPRINT = 1, the program will compute predicted values, probabilities, expected value of the error, and fixed effects for each observation/time period/equation analyzed. See RWRITE for details on what is printed.
- If RPRINT = 0 no residual analysis is done.

RWRITE = # where # is any non-negative integer. The default is 0. This option determines

if residual analysis information is punched/written on disk for each observation/time period/equation. This option can be used independently of RPRINT.

- If RWRITE = 0 nothing is punched.
- If # > 0 then # will be the device number that residual terms are punched/written on. Order of punch is the same as residual print, i.e., by equation, then time period, then cross-section observation.

For each equation/time period/observation the following four items are punched (or printed for RPRINT). (All computations are on a single equation basis.)

- For LOGIT Model Specifications:
  1. The dependent variable (0/1),
  2. The predicted $x'\beta$, i.e., $x_{it}'\hat{\beta} = x_{it}'b$,
  3. The probability $y_{it} = 1$, and
  4. A variable always 0.
- For PROBIT Model Specifications:
  1. The dependent variable (0/1),
  2. The predicted $x'\beta$, i.e., $x_{it}'\hat{\beta} = x_{it}'b$,
  3. The probability $y_{it} = 1$, and
  4. $E(\epsilon \mid y_{it}=1)$. (This variable can be used in other equations to correct for sample selection bias.)
- For Tobit Model Specifications:
  1. The dependent variable,
  2. The predicted latent dependent variable y^*, i.e., $\hat{y}_{it} = x_{it}'b$,
  3. The probability (PR) that y_{it} will not be at the masspoint, and
  4. $E(\epsilon \mid y_{it}$ is not at the masspoint), E. This number and PR allows the easy calculation of the expectation of the observed y variable given x, i.e. $E(y_{it} \mid x_{it})$ = (1 - PR) times the mass point value + PR times $(\hat{y}_{it} + E)$. The expectation of the observed y given it is not at a masspoint also follows easily as $\hat{y}_{it} + E$.
- For DBLTOB Model Specifications:
  1. The dependent variable,
  2. The predicted latent dependent variable y^*, i.e., $\hat{y}_{it} = x_{it}'b$,

3. The probability (PR1) that y_{it} will not be at either masspoint, and

4. $E(\epsilon \mid y_{it}$ is not at a masspoint), E. If a fixed effect is not used, this number and PR1 and PR2 allows the easy calculation of the expectation of the observed y variable given x, i.e. $E(y_{it} \mid x_{it})$ = PR2 times the non-zero mass point value + PR1 times $(\hat{y}_{it} + E)$. The expectation of the observed y given it is not at either masspoint also follows easily as $\hat{y}_{it} + E$.

5. The probability (PR2) that y_{it} will be at the non-zero masspoint. This item will not be printed/punched if a fixed effect is used for this equation. Note that the probability that y_{it} will be at the zero masspoint is 1 - PR1 - PR2.

- For TRUNCATE Model Specifications:

  1. The dependent variable,

  2. The predicted y before truncation, i.e., $\hat{y}_{it} = x_{it}'b$,

  3. The probability that y_{it} will be in the non-truncated range,

  4. $E(\epsilon \mid y_{it}$ is not truncated), E. The expection of y_{it} given it is not truncated follows as $\hat{y}_{it} + E$.

- For REGRES or IMPLICIT model Specifications;

  1. The dependent variable (0 for IMPLICIT),

  2. The predicted y, i.e., $\hat{y}_{it} = x_{it}'b$,

  3. The residual $y_{it} - x_{it}'b$, and

  4. The standardized residual $(y_{it} - x_{it}'b)/\sigma$,

where b is the estimate of β_0. If a fixed effect is computed for any equation, its value is punched/written as the fifth item, for a total format of (5D16.9). If some equations have fixed effects and some don't, all punches will be in (5D16.9) format, but the fifth variable will be zero for those equations without a fixed effect. If no equations have fixed effects, the above four items are punched/written the format is (4D16.9). If any equation has a DBLTOB dependent variable then the format used is (5D16.9) for all equations.

MWRITE = # where # is any non-negative integer. The default is 0. This option determines if the covariance matrix of model error terms is punched/written on disk.

- If MWRITE = 0 nothing is punched.

- If # > 0 then # will be the device number that the covariance matrix of model error terms will be punched/written on. This option only applies if TYPE = 4, and is useful an input to the LISREL program if dichotomous or Tobit indicators are present. The matrix

is punched as a single vector, lower-triangular matrix in (5D16.9) format. Element order is (1,1), (2,1), (2,2), (3,1), (3,2), (3,3), (4,1), ...

3.8. TIME VARIATION CARD

This set of commands is optional; it is used if and only if TIME > 1. It is used to identify any variables which are constant over time, or any variables which are to be expressed as "deviations-about-cross-sectional-means". For the former type, list any variables which are constant over time for each cross-sectional unit (thus all variation in these variables is cross-sectional) on this card. Each variable should be listed as: V# = C where # is the number of the variable. Variable names can be substituted for numbers. Names do not have to be bracketed (its optional here). The "=C" term must be used for the first variable on the card, but is optional for all others. Variable strings can also be used, indicated by a "-". Thus, $V\#_1$ - $V\#_2$ = C will imply that all variables from $\#_1$ to $\#_2$ are constant over time.

If variables (including those in strings) are preceded by an "N" they will not be assumed to vary. Thus, for example V1-V10=C,NV8 states that all variables between one and ten except eight are constant. Variable names can be used in strings (its number is used in determining the range) and with the N term. In the latter case variable names must be bracketed, i.e. N[REEDER]. Variables (or strings) should be separated by blanks or commas and $ used for line continuation. If a variable is not listed it is assumed to vary over time. This option saves storage space and helps to specify inter-temporal orthogonality conditions if TYPE = 3. Only those variables used as independent variables in the run need be specified.

This card can also be used to specify variables which will be used in the form of "deviations-about-cross-sectional-means." Each such variable should be listed as: V# = D where # is the number of the variable. Any such variables must use = D (unlike constant variables where only the first must have = C). The "N" option can also be used, as well as strings and variable names. Variables can be mixed, with some =C and some =D. A variable cannot be used both ways however. All variables assigned this option will be used in deviation form except dependent variables used for logits/probits/tobits/truncated/double tobits and variables used for likelihood weighting. For regression dependent variables, only observations used for an equation will be used in computing the cross-sectional mean. The use of this option is equivalent to using a fixed effect model (FIXED = 1) for a regression, and should generally not be used with either a fixed or random effects model (TYPE = 2).

If no variables are specified to be constant over time or deviations about their cross-sectional means, then do not include the TIME VARIATION CARD.

3.9. BASIC EQUATION CARD

This card, the BASIC EQUATION CARD, followed by the EQUATION VARIABLE CARD (optional) and EQUATION TESTING CARD (optional), must be read in for each equation (EQUA) in order. That is, the cards (possibly 3 if both optional cards are included) for equation one are specified first, followed by the cards for equation two, followed by the cards for equation three, etc....

Option	Description

DEP = $V\#_1$ *or* DEP = $V\#_1$.XX.$V\#_2$ *or* DEP = $V\#_1$.XX.$\#_3$
where $\#_1$ and $\#_2$ are any integers from 1 to VAR*, $\#_3$ is any real number not expressed in scientific notation and XX is one of the logical operators (GT, LT, GE, LE, EQ, NE). Variable names bracketed by [and] can be substituted for $V\#_1$ and/or $V\#_2$. Only DEP = $V\#_1$ (or named) is necessary with defaults for the rest explained below. This option generally sets the equation's dependent variable as $V\#_1$.

- For MODEL = REGRES that is all that is necessary.
- If MODEL = IMPLICIT then DEP should not be included.
- If MODEL = PROBIT or MODEL = LOGIT then the event occurring Y = 1) is determined by whether the expression following DEP = is true (using standard FORTRAN treatment of the operator). If true, then Y = 1, otherwise Y = 0. The value in $V\#_1$ can be compared to either another variable, $V\#_2$, or a real number, $\#_3$.
- If MODEL = TOBIT then the observation is a non-masspoint with value of the variable $V\#_1$ if the expression following DEP = is true. If it is false then a masspoint with value of the variable $V\#_2$ or $\#_3$ is assumed. Thus lower or upper bounds and/or variable masspoints are allowed. Only the operators LT, GT, LE, or GE can be used.
- If MODEL = DBLTOB, then one of the masspoints is set by DEP as with MODEL = TOBIT. The other masspoint is 0. If $V\#_1$.LT. or .LE. is used, then $V\#_2$ (or $V\#_3$) is the upper bound and 0 is the lower bound. If $V\#_1$.GT. or .GE. is used then $V\#_2$ (or $V\#_3$) is the lower bound and 0 is the upper bound.
- If MODEL = TRUNCATE then the expression determines which observations are used for the equation with $V\#_2$ or $\#_3$ as the truncation point.

If only $V\#_1$ is specified with LOGIT, TOBIT, PROBIT or TRUNCATE, then the default is .GT.0

FIXED = # where # is 0 or 1. The default is 0. This option applies only with panel data (TIME > 1) and determines if an additive fixed effect is to be included in the equation.

- If FIXED = 0, no fixed effect is included in the equation.
- If FIXED = 1 the program will estimate a fixed effect, which is entered additively into the equation, for each unit observation if TYPE = 1 or 3. This is equivalent to estimating a separate intercept term for each cross-sectional unit. Thus, an equation constant should not generally be specified. Therefore, the command CONST is automatically set to one for all equations where FIXED = 1. With a LOGIT or PROBIT model, any observations with a constant dependent variable over time will not be used in estimation. Similarly, for Tobit models any observations located at one masspoint for all time periods will be excluded. The final fixed effect can be printed out/punched

using RWRITE or RPRINT.

The fixed effect is calculated recursively, and independently of the other parameters. At each iteration, for each cross-section unit, the program calculates the intercept which maximizes the likelihood for the equation. This is usually equivalent to finding the intercept which sums the residuals to zero for the equation (hence the reason not to have an equation constant). The program uses a one-term Newton Raphson iteration scheme to solve iteratively for each intercept allowing up to ten iterations for convergence by the criteria FETOL. With regressions this should only take one iteration. The computed intercept is used as a starting value for the next iteration. If problems are encountered the program will stop and may require different starting values (see FESTAR below). Several words of warning are in order. Since a separate intercept is estimated for each cross-section unit, no independent variables should be used which do not vary over time. A sign that this has been violated is the failure of the program to invert the second derivatives for standard errors satisfactorily. Also note that when a fixed effect is used, the program's standard errors--which ignore the fixed effect--may be in error. Note as well, that for regression models, parameters equivalent to a fixed effect model can be computed by using the TIME VARIATION CARD to express the independent and dependent variables as deviation-about-the-means. This method is cheaper than FIXED = 1.

FESTAR = # where # is any real number not expressed in scientific notation. The default is 0. This option applies only with a fixed effect (FIXED = 1) and determines the starting value of the fixed effect for all observations in the first iteration. Since the estimated fixed effects are "intercepts" for each cross-sectional unit, the value of FESTAR should be set to the "average" intercept. A good estimate would be the constant term from a previous equivalent model run without a fixed effect. If observations are scaled to have an average intercept of approximately zero the default can be used. Note, that if a poor starting value is used the program may not be able to iterate at all and will "bomb" on the first iteration.

UCROS = $\#_1$ TO $\#_2$
where $\#_2$ is any integer from 1 to OBSV and $\#_2$ is any integer from $\#_1$ to OBSV. The default for $\#_1$ is 1 and for $\#_2$ is OBSV. If UCROS is not specified here, the default is to use all observations specified on the MASTER CONTROL CARD. This option determines which observations are used for this equation. Instructions are the same as on the MASTER CONTROL CARD. The observations used will be a subset of those specified on the MASTER CONTROL CARD.

XCROS = $\#_1$ TO $\#_2$
where $\#_2$ is any integer from 1 to OBSV and $\#_2$ is any integer from $\#_1$ to OBSV. If XCROS is not specified here, the default is to use all observations specified on the MASTER CONTROL CARD. This option determines which observations are excluded for this equation. It cannot be used with UCROS on the BASIC EQUATION CARD. Instructions are similar to those of UCROS.

UOBSV = V$\#_1$.XX.V$\#_2$ *or* UOBSV = V$\#_1$.XX.$\#_3$
where $\#_1$ and $\#_2$ are any integers from one to VAR*, $\#_3$ is any real number

not expressed in scientific notation and XX is one of the logical operators (GT, LT, GE, LE, EQ, NE). Variable names bracketed by [and] can be substituted for $V\#_1$ and/or $V\#_2$. The default if UOBSV (and UCROS and XCROS) is not specified is to use all observations/time periods specified on the MASTER CONTROL CARD. Same general instructions are used as UCROS. This option determines which specific observations/time periods are used for this equation only.

MODEL = XX where XX equals LOGIT, PROBIT, TOBIT, DBLTOB, TRUNCATE, IMPLICIT and REGRES. This option overrides the model specification on the MASTER CONTROL CARD for this equation only. If not specified on the BASIC EQUATION CARD, the program defaults to the model specification given on the MASTER CONTROL CARD.

SIGMA = # where # is 0, 1, 2, 3, or 4. The default is to set SIGMA to the value set on the MASTER CONTROL CARD. This option overrides the treatment of the model error term variance/standard deviation given on the MASTER CONTROL CARD for this equation only.

3.10. EQUATION VARIABLE CARD

This card is included for each equation, immediately following its corresponding BASIC EQUATION CARD if linear equations are used (NONLIN = 0). If NONLIN > 0, the card is read only if TYPE = 3. With linear systems, this card is used to specify the independent variables used in each equation with the exception of the constant which is automatically included if CONST = 0. If CONST = 1 (or FIXED = 1) the constant may be selectively included as variable V0 or CONSTANT (it can't be included in strings). Variables are listed as $V\#_1 = \#_2$, where $\#_1$ is the number of the variable, and $\#_2$ is an optional code. Variable names can be substituted for $V\#_1$. Unlike most other cards, names do not have to be bracketed by [and] here (its optional).

Variable strings can also be used, indicated by a "-" between variables. Thus $V\#_1 - V\#_2 = \#_3$ (the $= \#_3$ is optional) implies that all variables from $\#_1$ to $\#_2$ will be included in the equation (or given the code $\#_3$). If variables (including those in strings) are preceded by an "N", they will not be included. Thus, for example, V1-V10,NV8 implies that all variables from one to ten except variable eight will be included. If CONST = 0, the constant may be selectively excluded from equation by the use of NV0 or N[CONSTANT].

Variable names can be used in strings (the number is used to determine the string range) and with the N command. When names are used after N they must be bracketed by [and]. If a variable appears more than once on the EQUATION VARIABLE CARD its status will be determined by the last reference. Variables (or strings) must be separated by blanks or a comma.

Variables can be placed in any order on the EQUATION VARIABLE CARD. However, the program will always reorder them according to their variable numbers. Variable coefficients will be ordered in ascending order according to their numbers, followed by the constant (if used) for each equation.

The variable code, $\#_2$, has fifteen options.

- If $\#_2$ = 0 (the default if = $\#_2$ not included) the variable is included in the equation with contemporaneous orthogonality conditions only. This option (or =15) should be used unless TYPE = 3. If TYPE = 3, this option is indicated by a "C" next to the variable on printout.

- If $\#_2$ = 1 the variable is included in the equation with full contemporaneous, forward (lead), and backward (lag) orthogonality conditions with the number of leads and lags determined by LAGS. This is equivalent to option (=0) if TIME = 1, TYPE is not equal to 3, or the variable does not vary over time. This option is indicated by a "A" next to the variable on printout.

- If $\#_2$ = 2 the variable is included in the equation with forward orthogonality conditions (residual is orthogonal to future and contemporaneous values of the variable), with the number determined by LAGS. This is equivalent to option (=0) if TIME = 1, TYPE is not equal to 3, or the variable does not vary over time. This option is indicated by a "F" next to the variable on printout.

- If $\#_2$ = 3 the variable is included in the equation with backward orthogonality conditions (residual is orthogonal to past and contemporaneous values of the variable), with the number determined by LAGS. This is equivalent to option (=0) if TIME = 1, TYPE is not equal to 3, or the variable does not vary over time. This option is indicated by a "B" next to the variable on printouts.

- If $\#_2$ = 4 the variable is included in the equation but is not used in any orthogonality conditions. This is useful with endogenous variables, but restrictions in the parameters or extra instruments are generally needed.

- If $\#_2$ = 5 the variable is included in the equation but contemporaneous orthogonality conditions are not imposed. Forward and backward (lead and lag) orthogonality conditions are imposed however. This is the same as option (=4) unless TYPE = 3 and TIME > 1.

- If $\#_2$ = 6 the variable is included in the equation but contemporaneous orthogonality conditions are not imposed. Forward only conditions are imposed (future). This is the same as option (=4) unless TYPE = 3 and TIME > 1.

- If $\#_2$ = 7 the variable is included in the equation but contemporaneous orthogonality conditions are not imposed. Backward only conditions are imposed (past). This is the same as option (=4) unless TYPE = 3 and TIME > 1.

- If $\#_2$ = 8 same as option (=0) except the variable is not included in the equation.

- If $\#_2$ = 9 same as option (=1) except the variable is not included in the equation.

- If $\#_2$ = 10 same as option (=2) except the variable is not included in the equation.

- If $\#_2$ = 11 same as option (=3) except the variable is not included in the equation.

- If $\#_2$ = 12 same as option (=5) except the variable is not included in the equation.

- If $\#_2$ = 13 same as option (=6) except the variable is not included in the equation.
- If $\#_2$ = 14 same as option (=7) except the variable is not included in the equation.
- If $\#_2$ = 15 the variable is not included in the equations nor is it in any orthogonality conditions. This option is equivalent to not specifying the variable or including NV$\#_1$.

3.11. EQUATION TESTING VARIABLE CARD

This card is read only if TEXOG = 1 (exogeneity tests) and TYPE = 3. If TEXOG = 1 this card must be present even if it is blank. This card should immediately follow the EQUATION VARIABLE CARD for each equation. Any variables for which exogeneity restrictions are tested should be specified. The format is identical to the EQUATION VARIABLE CARD. Users may (optional) precede the variable code by the character "T" (for identification only, e.g., V22 = T1). Only tested restrictions need be listed, thus only those variables with more orthogonality conditions than in the estimating equations need be specified. No variables can be used with less orthogonality restrictions than in the estimating equation. Furthermore, the number of implied estimated parameters cannot be larger than those in the estimating equation. Thus all new variables must have codes of 8-14. If the EQUATION TESTING VARIABLE CARD is blank, any over-identifying restrictions in the basic model will be tested.

3.12. ORTHOGONALITY CONDITION TEST CARD(S)

This card is read TSUB times if TSUB > 0. On each card, numbers are read specifying the orthogonality conditions to be tested with a chi-square test as a subset of the large exogeneity test specified on the EQUATION TESTING VARIABLE CARD. Each chi-square test has a separate card. Cards have the following form:

SUBSET = $\#_1$, $\#_2$,..., $\#_N$

where $\#_1$,...,$\#_N$ are the numbers of the tested restrictions in the sub-test. N cannot be larger than the number of over-identifying orthogonality conditions in the model (including those specified on the EQUATION TESTING VARIABLE CARD). $\#_1$,...,$\#_N$ refer to the order of the orthogonality condition and can be computed as follows. Variables are entered into the model first by equation then by ascending order given their variable number with the constant last (V1,V2,,...,V0). Within variable, the order is:

1. Contemporaneous ORTHOGONALITY CONDITION;
2. lead one period ORTHOGONALITY CONDITION;
3. lag one period ORTHOGONALITY CONDITION;

.
.
.

n. lead TLAG period ORTHOGONALITY CONDITION;
n+1. lag TLAG period ORTHOGONALITY CONDITION.

The orthogonality condition for SIGMA (not used if SIGMA = 3,4) follows the variables for regression, implicit function, Tobit, double Tobit, and truncated models for each equation, i.e., equation 1 variables, sigma 1, equation 2 variables ....

3.13. NON-LINEAR EQUATION CARD(S)

Each equation specified by the system must have a separate equation card input if NONLIN = 1 or 2. If NONLIN = 1, cards must also be input for each equation error variance (standard deviation) estimated (exclude probits/logits), the random effects parameter for each equation if TYPE = 2, and each cross-equation correlation if TYPE = 4. Given that there are EQUA total equations to be estimated, the order of equation cards is:

Equation card for equation 1.

Error Variance (or Standard Deviation) for equation 1 (only NONLIN=1).

Random effects for equation 1 (only NONLIN=1).

Equation card for equation 2.

Error Variance (or Standard Deviation) for equation 2 (only NONLIN=1).

Random effects for equation 2 (only NONLIN=1).

.

.

.

Equation card for LAST equation.

Error Variance (or Standard Deviation) for LAST equation (only NONLIN=1).

Random effects for LAST equation (only NONLIN=1).

Correlation (or Covariance) between equations 1 and 2 disturbances (only NONLIN = 1).

Correlation (or Covariance) between equations 1 and 3 disturbances (only NONLIN = 1).

Correlation (or Covariance) between equations 2 and 3 disturbances (only NONLIN = 1).

Correlation (or Covariance) between equations 2 and 4 disturbances (only NONLIN = 1).

.

.

.

Correlation (or Covariance) between SECOND-TO-LAST and LAST equation

disturbances (only NONLIN = 1).

Note that only equation cards are included if NONLIN = 2.

Equation cards must have the following form. The first character is an equation delimiter. For an equation it must be an "E". For an error variance it can be an "S" signifying that the estimated parameter is the error standard deviation or an "V" signifying that it is the variance. This overrides what is given in the SIGMA option (if however, SIGMA = 3 or 4 and TYPE = 3 then the sigma orthogonality condition is not imposed, whether or not the variance or standard deviation is specified is determined here). For a random effects card it must be a "P". For a cross-equation correlation it must be a "P" indicating the parameter is the cross-equation correlation or a "C" indicating that it is the cross-equation covariance.

The equation delimiter may be followed (optional) by a number indicating equation number (although equations must be read in order). This number need not be specified. The next character must be "=". After "=" the equation itself is spelled out. Each equation is made up of functions of variables (indicated by V# or names bracketed by [and]), parameters (indicated by B#), and numbers (#, can be real). The parameters can be numbered in any order, but all parameters from B1-B(max) must be specified on some system card. If NONLIN = 2 then equation error variances and intercorrelations (optional) will be assigned numbers following B(max).

Each equation is made up of "elements" combined by the operations +, -, *, /. Each "element" has the following form:

a * (or /) b ** c **(i.e. ab^c)**

where:

a = # (any real number not expressed in scientific notation), *and*

b = # (any real number not expressed in scientific notation), *or*

b = LN# (natural log of any positive number), *or*

b = V# (any variable. The name of the variable bracketed by [and] may be substituted for V#), *or*

b = LNV# (natural log of any variable. The name of the variable bracketed by [and] may be substituted for V#), *or*

b = B# (any parameter), *or*

b = LNB# (natural log of any parameter), *or*

b = (...) where the ... are other elements with a single layer of parenthesis, *or*

b = LN(...) natural log of what is in the parenthesis, *and*

c = # (any real number not expressed in scientific notation. There is a restriction

however. If "c" is larger in absolute value than 1, it will be rounded to the nearest integer), *or*

c = V# (any variable. The name of the variable bracketed by [and] may be substituted for V#), *or*

c = B# (any parameter).

Note: only one layer of parenthesis is allowed and parenthesis cannot follow exponentiation (**). Any of the terms **a, b,** c are optional.

Some examples are:

```
E=B1+B2*V2+B1*(B3**2+LNV6/B6)**V3
S3=B2**-B1-6.37+LN(B3/B1/B7*B2+3)*(B1+B2)
E=B1+[REEDER]**2*B2+([NEW]*B3+[REEDER]*V5*B4)**V2
P=B1*B2*([REEDER]+LN[NEW])+B2*[DEPTOB]
```

3.14. RESTRICTION CARD(S)

Each restriction specified by the system must have a separate restriction card if RESTRI > 0. For each parameter restriction, a restriction equation is read in on a separate line. The restriction equations have the following form. On the left-hand side of the expression, one must specify the parameter or a function of the parameter to be restricted. In particular, the left-hand side of the restriction can take the following form:

a * (or /) b ** c = **(i.e. ab^c)**

where:

a = # (any real number not expressed in scientific notation), *and*

b = B# (the parameter to be restricted), *and*

c = # (any real number not expressed in scientific notation. There is a restriction however. If "c" is larger in absolute value than 1, it will be rounded to the nearest integer).

a and c are optional. **b** is not. Note that a parameter can be restricted only once.

Each restricted parameter (or function of a restricted parameter) can be restricted to be equal to functions of constants and/or unrestricted parameters. In particular the right-hand side of the restriction expression can take the following form:

a * (or /) b ** c (or -c) **(i.e. ab^c)**

where:

a = # (any real number not expressed in scientific notation), *and*

b = # (any real number not expressed in scientific notation), *or*

b = B# (any unrestricted parameter), *or*

b = (...) where the ... are other elements with a single layer of parenthesis, *and*

c = # (any real number not expressed in scientific notation. There is a restriction however. If "c" is larger in absolute value than 1, it will be rounded to the nearest integer).

Note: In the right-hand terms of the restriction, only one layer of parenthesis is allowed and parenthesis cannot follow exponentiation (**). Any of the terms **a, b, c** are optional.

In the specification of restrictions, the parameters are defined in the following way. If non-linear equations are used, then the parameters are specified using the same "B" numbers used to write out the equations on the NON-LINEAR EQUATION CARD(S). For NONLIN = 2, error variance and intercorrelation parameters are numbered following the maximum parameter used in the non-linear equations. The numbering order is: the error sigma for equation one, the random effect of equation one (if used), etc. These are followed by equation error covariances using the same pattern as NONLIN = 0 if TYPE = 4. If linear systems are used (NONLIN = 0), all parameters are specified as B#, where the # denotes the order of the parameter. This parameter order is as follows. The coefficients corresponding to the variables in Equation One come *first*, with the coefficient number being equal to the ascending order of the variable numbers (i.e. V1,V2, ...) except for the constant (V0) which always is the last coefficient in each equation. Then comes the error sigma (optional) of Equation One, the random effect of Equation One (optional), then the same parameters repeated for Equation Two, etc. If TYPE=4 these are followed by the correlation of equations 1 and 2, 1 and 3, 2 and 3, 1 and 4, ...

Some examples of restriction cards are:

```
2.73/B2=B4/B5+B7**4
-B2**.5=B1*(B4+B5)**2+B6/B5**.5
```

3.15. COEFFICIENT STARTING VALUE FORMAT CARD

This card is read only if FORMAT=1 on the MASTER CONTROL CARD and START = 0, 1, 3 (or if ECORR =3 or TCORR =3). The FORTRAN format used to read coefficient values on the COEFFICIENT STARTING VALUE CARD is read here. It should be a standard FORTRAN format (with parenthesis) but not take more than one line. For example, (4F10.4,3X,3F12.3)

3.16. COEFFICIENT STARTING VALUE CARD

If START = 0, 1, 3 (or ECORR = 3 or TCORR = 3), the parameter starting values should be read here by either the format of (5D16.9) if FORMAT=0 or the format read in on the preceding FORMAT CARD. Parameter order is: equation one variable coefficients (in ascending order of numbers but with constant last), equation one error sigma (optional), equation one random effects (optional), equation two parameters etc....., correlation of equations 1 and 2 (optional if TYPE=4), correlation of equations 1 and 3 (optional), correlation of equations 2 and 3 (optional) .... If NONLIN = 1 then parameter order corresponds to the B's specified on the NON-LINEAR EQUATION CARD(S). If NONLIN = 2 then parameters of the non-linear equations following the order given by the B's on the NON-LINEAR EQUATION CARD(S). These are followed by error sigmas, random effect parameters, and equation error covariances in the same order as NONLIN = 0. If NONLIN > 0 then starting values equal to zero should be avoided (unless restricted) as they can cause problems.

3.17. WEIGHTING MATRIX COEFFICIENT FORMAT CARD

This card is read only if FORMAT=1 on the MASTER CONTROL CARD and START = 1 (or START = 1, 3 and if ECORR =3 or TCORR =3). The FORTRAN format used to read coefficient values on the WEIGHTING MATRIX COEFFICIENT CARD is read here. It should be a standard FORTRAN format (with parenthesis) but not take more than one line.

3.18. WEIGHTING MATRIX COEFFICIENT CARD

If START = 1 (or START = 1, 3 and if ECORR = 3 or TCORR = 3), the coefficients used to form the weighting matrix if TYPE = 3 and OPTWT = 0 should be read here by either the format of (5D16.9) if FORMAT=0 or the format read in on the preceding FORMAT CARD. Parameter order is the same as on the COEFFICIENT STARTING VALUE CARD.

3.19. TIME PERIOD CORRELATION FORMAT CARD

This card is read only if FORMAT=1 on the MASTER CONTROL CARD and TCORR = 1 (or TCORR = 3 and START = 0, 1). The FORTRAN format used to read starting values of time-period correlations on the TIME PERIOD CORRELATION CARD is read here. It should be a standard FORTRAN format (with parenthesis) but not take more than one line.

3.20. TIME PERIOD CORRELATION CARD

If TCORR = 1 (or TCORR = 3 and START = 0, 1), starting values for the time period correlations are read by either the format of (5D16.9) if FORMAT=0 or the format read in on the preceding FORMAT CARD. The time period correlations are read as elements of a continuous lower triangular matrix (i.e. correlation 1 and 2, correlation 1 and 3, correlation 2 and 3, etc...)

3.21. EQUATION CORRELATION FORMAT CARD

This card is read only if FORMAT=1 on the MASTER CONTROL CARD and ECORR = 1 (or ECORR = 3 and START = 0, 1). The FORTRAN format used to read starting values of cross-equation correlations on the EQUATION CORRELATION CARD is read here. It should be a standard FORTRAN format (with parenthesis) but not take more than one line.

3.22. EQUATION CORRELATION CARD

If ECORR = 1 (or ECORR = 3 and START = 0, 1), starting values for the equation correlations are read by either the format of (5D16.9) if FORMAT=0 or the format read in on the preceding FORMAT CARD. The cross-equation correlations are read as elements of a continuous lower triangular matrix (i.e. correlation 1 and 2, correlation 1 and 3, correlation 2 and 3, etc...)

3.23. TRANSITION CARD

This card signals the end of the main **HotzTran** program and the desire to read another data file. It has the following commands

Option	Description
SAMPLE	This command instructs the program to return to the beginning of the program to read another DATA CONTROL CARD. If SUBROUTINES USER or USEO are used, the variable ISAMP will indicate the number of the sample being read. It can be accessed by including the statement: COMMON /G/ISAMP in the subroutine. The entire command SAMPLE must be spelled out (no abbreviations).
REWIND = #	where # is any positive integer. This option will rewind the file on device #. The entire command REWIND must be spelled out (no abbreviations).
ENDFILE = #	where # is any positive integer. This option will end file the file on device #. The entire command ENDFILE must be spelled out (no abbreviations).

3.24. SUMMARY OF PROGRAM OPTIONS

By way of summary, the following is an alphabetical list of Options which may be specified on the DATA CONTROL, MASTER CONTROL and BASIC EQUATION CARD. For each option, the following information is provided: (1) the card on which the option appears, (2) a brief description of the option, (3) the values the option can take, and (4) the page in the manual where the option is described in detail. The default value for each option is either underlined or noted as "(def. XXXX)."

Option	Card	Description of Option	Values	Page
			(0 generally means option not used)	
CNVRG	MASTER	Convergence criteria control	0, 1, 2, 3, 4	42
CONST	MASTER	Constant control	0, 1	32
CWRITE	MASTER	Coefficient punch/disk write control (0 = no punch)	0, 1, 2,	44
DEP	BASIC	Dependent variable number	V#	48
DEP	BASIC	Determination of dependent variable (XX = GT, etc.)	$V\#_1$.XX.$V\#_2$ or $\#_3$	48
DISK	DATA	Internal observation disk control	0, 1, 2	22
DISK	MASTER	Internal model run data disk control	0, 1	38
ECORR	MASTER	Computation of inter-equation correlations (if TYPE = 1, 3)	0, 1, 2, 3	34
EQUA	MASTER	Number of equations	1, > 1	29
ESTIMA	MASTER	Estimation procedure	0, 1, 2, 3, 4, 5,	42
FESTAR	BASIC	Fixed effect starting value	0., #	49
FETOL	MASTER	Tolerance for fixed effect convergence	> 0 (def. .00001)	43
FIXED	BASIC	Fixed effect control (if TYPE = 1, 3)	0, 1	48
FORMAT	DATA	Data input format	0, (...), USER, USEO	21

FORMAT	MASTER	Format control for starting values	0, 1	41
FTOL	MASTER	Tolerance for function change convergence	> 0 (def. .0000001)	43
GTOL	MASTER	Tolerance for gradient convergence	> 0 (def. .00001)	43
HERMIT	MASTER	Number of hermite points (if TYPE = 2)	1-4, 6, 9	33
INPUT	DATA	Input device (FORTRAN) number	INP, > 0	22
IPRINT	MASTER	Iteration print control (1 = no print)	0, 1	44
ITERA	MASTER	Maximum number of iterations	0, >0 (def. 50)	43
LAGS	MASTER	Number of leads/lags for instruments (if TYPE = 3)	0 - TIME-1	37
LINE	DATA	Line width for printout (1 = 80 characters/line)	0, 1	22
MAVER	MASTER	Number of moving average sums in std. error calculations	0 - TIME-1	36
MISS	DATA	Missing variable value code	<#, =#, ># (def. no miss)	23
MISS	MASTER	Equation missing value control	0, 1, 2	38
MODEL	BASIC	Model-type	LOGIT, PROBIT, TOBIT, DBLTOB, TRUNCATE, REGRES, IMPLICIT, (def. MASTER)	50
MODEL	MASTER	Default model-type for all equations	LOGIT, PROBIT TOBIT, DBLTOB, TRUNCATE, REGRES, IMPLICIT	29
MWRITE	MASTER	Device Number (FORTRAN) of equation error covariance punch (if TYPE = 4)	0, > 0	46
NAME	DATA	Variable name control	0, 1	23

NONLIN	MASTER	non-linear equation control	0, 1, 2	31
OBSV	DATA	Number of cross-sectional units	0, > 0	20
OPTWT	MASTER	Orthogonality weighting scheme (if TYPE = 3)	0, 1, 2	36
OUTLY	MASTER	Outlier control	0, 1, 2	42
PCOVA	MASTER	Print control for parameter covariance	0, 1	44
PRINT	DATA	General data set print control	0, 1, 2, 3, 4, 5, 6, 7	22
PSTAN	MASTER	Standard error control	0, 1, 2, 3	35
PTOL	MASTER	Tolerance for parameter change convergence	> 0 (def. .0001)	43
RANDOM	MASTER	Factor loading control (if TYPE = 2)	0, 1, 2	33
RESTRI	MASTER	Number of parameter restrictions	0, > 0	32
RHO	MASTER	Covariance/correlation control (if TYPE = 4)	0, 1	35
RPRINT	MASTER	Residual analysis print control	0, 1	44
RWRITE	MASTER	Device number (FORTRAN) for residual punch	0, > 0	44
SCALE	MASTER	Coefficient scaling control (1 = no scale)	0, 1	41
SECS	MASTER	Maximum number of CPU seconds	0, >0 (def. 5)	44
SIGMA	BASIC	Model error variance control	0, 1, 2, 3, 4 (def. MASTER)	50
SIGMA	MASTER	Default model error variance control	0, 1, 2, 3, 4	34
START	MASTER	Starting value control	0, 1, 2, 3, 4	40
STOP	MASTER	Termination command		29
TCORR	MASTER	Computation of inter-time correlations (if TYPE = 1, 3)	0, 1, 2, 3	33

TEXOG	MASTER	Exogeneity test control (if TYPE = 3)	0, 1	37
TIME	DATA	Number of time periods or variable	1, > 1 or V#	21
TLAG	MASTER	Number of test lead/lags (if TEXOG = 1)	LAGS - TIME-1	38
TRANS	DATA	Number of data transformations	0, > 0	24
TSUB	MASTER	Number of chi-square tests (if TEXOG = 1)	0, > 0	38
TYPE	MASTER	Estimation procedure	1, 2, 3, 4	30
UCROS	BASIC	Cross-sectional unit control	$\#_1$ TO $\#_2$	39
UCROS	MASTER	Cross-sectional unit control (all equations)	$\#_1$ TO $\#_2$ (def. all)	49
UOBSV	BASIC	Observation specific control (XX = GT, etc.)	$V\#_1$.XX.$V\#_2$ or $\#_3$	49
UOBSV	MASTER	Observation specific control (all equations, XX = .GT, etc.)	$V\#_1$.XX.$V\#_2$ or $\#_3$ (def. all)	39
UTIME	MASTER	Time period control (if TIME > 1)	$\#_1$ TO $\#_2$ (def. all)	39
VARI	DATA	Number of Variables read	> 0	20
WEIGHT	MASTER	Variable number for weighting	V#	32
XCROS	BASIC	Cross-sectional unit exclusion	$\#_1$ TO $\#_2$	49
XCROS	MASTER	Cross-sectional unit exclusion (all equations)	$\#_1$ TO $\#_2$ (def. none)	39
XTIME	MASTER	Time period exclusion (if TIME > 1)	$\#_1$ TO $\#_2$ (def. none)	39

3.25. SUBROUTINES USER AND USEO

There are two user-supplied subroutine which can be expanded by users to read in observations one-at-a-time (SUBROUTINE USER) or as an entire observation matrix (SUBROUTINE USEO). These subroutines are structured as follows:

```
C     SUBROUTINE USER
C     X IS THE OBSERVATION VECTOR TO BE FILLED, - NVAR IS THE NUMBER OF
C     VARIABLES, - INUM IS THE NUMBER OF THE OBSERVATION ( INCLUDING
C     EXCLUDED OBSERVATIONS), - INN IS THE INPUT DEVICE, - IDD IS A
C     RETURN CODE  0=PROCESS AND CONTINUE (DEFAULT), 1=DON'T PROCESS
C     BUT CONTINUE, 2=PROCESS THIS OBSERVATION THEN STOP, 3=DON'T PROCESS
C     THIS OBSERVATION AND STOP, 4=ERROR IN INPUT TERMINATE PROGRAM.

      SUBROUTINE USER(X,NVAR,INUM,INN,IDD)
      DIMENSION X(1)
C
C     A READ STATEMENT AND ANY TRANSFORMATIONS AND/OR
C     VARIABLE CREATION GOES HERE USING STANDARD FORTRAN
C
      RETURN
      END


C     SUBROUTINE USEO
C     DATA IS THE ENTIRE OBSERVATION MATRIX. IT IS DIMENSIONED LOBS
C     BY NVAR, WHERE NVAR IS # VARIABLES READ (EXCL. CONSTANT) AND LOBS
C     IS THE OBSERVATION DIMENSION. NOBS IS THE NUMBER OF OBSERVATIONS.
C     IF READ TO END-OF-FILE SET NOBS TO RIGHT AMOUNT ON RETURN. FILL
C     ALL ROWS OF DATA UP TO NOBS. INN IS THE INPUT DEVICE. SET IDD=1
C     IF INPUT ERROR.
      SUBROUTINE USEO(DATA,LOBS,NVAR,NOBS,INN,IDD)
      DIMENSION DATA(LOBS,NVAR)
C
C     A READ STATEMENT AND ANY TRANSFORMATIONS AND/OR
C     VARIABLE CREATION GOES HERE USING STANDARD FORTRAN
C
      RETURN
      END
```

4. INSTRUMENTAL VARIABLES (IV) AND SPECIALTY MODEL ESTIMATION

The objective of this Chapter is to describe the form and properties of the major estimators used in **HotzTran**, and illustrate several specialty models. Many of these estimators, such as maximum likelihood estimators, are well known. We also heavily utilize a type of estimator which may not be as familiar. These estimators reside within a class of generalized method of moments (GMM) estimators studied by Hansen (1982) and can be represented in the same form as the nonlinear instrumental variables (IV) estimator of Amemiya (1974, 1977) and Jorgenson and Laffont (1974). These estimators will also be shown to be related to nonlinear generalized least squares (GLS) estimators. These estimators are Nonlinear Instrumental Variables (IV) estimators or Orthogonality Condition (OC) estimators.

We present a discussion of estimation utilizing OC or IV estimators using a multi-time period probit model as an example in the first section. We describe the asymptotic properties of these estimators and also discuss a number of tests which can be performed and for which the program can supply test statistics. Included in these tests is an "exogeneity" test of variables which might be used as instrumental variables. In the next section, we discuss how to implement these procedures in **HotzTran**. We show how these techniques can be applied to non-probit and multiple-equation systems as well. In the third section we discuss the specific implementation of IV methods in **HotzTran**. In the fourth section we show how these methods can be applied to other model forms such as Tobits, regressions and multiple equation systems. We conclude the chapter with three sections detailing specialty models which may be used quite often with **HotzTran**. These include: (1) the sample selection bias model, (2) the Multiple-Cause Multiple-Indicator (MIMIC) model, and (3) non-linear regression models.

4.1. IV Estimation with the multi-period Probit Model

In this section, to make our presentation more concrete, we focus our discussion on the estimation of the multiperiod probit model. However, the estimation form can be equally well applied to any of the other model forms available in **HotzTran** (e.g. Tobit, regressions, etc.). The principles utilized in multi-time period models can also be shown to apply to multi-equation estimation. Extensions to these model forms are presented in the final section of this chapter.

Throughout our discussion here we assume a multiperiod probit model as expressed in equations (2.1.4) and (2.1.5). However, it is necessary to be more specific about the nature of the stochastic environment generating the δ's and the x's. Although this strong an assumption is not always necessary, throughout this section we maintain the assumption that δ_{it} is uncorrelated with $x_i = \{x_{i1}', x_{i2}', \ldots, x_{it}', \ldots, x_{iT}'\}$, i.e. functions of current, past, and future x's; thus that the x's are strictly exogenous. In the spirit of most cross-sectional analyses, we also rule out cross-sectional dependencies by assuming that the random vectors $\{(\delta_i', x_i')\}_{i=1}^{N}$, where $\delta_i' = (\delta_{i1}, \ldots, \delta_{iT})$, are independent and identically distributed. We maintain the standard probit assumption that the δ's are standard normally distributed, but allow for the disturbances of a given individual to be serially correlated. That is, $E(\delta_{it}\delta_{i\tau})$ need not be equal to zero for $t \neq \tau$. Given these assumptions, there are a number of possible estimation strategies.

Maximum likelihood is an obvious strategy for estimation. However, as discussed in Avery, Hansen, and Hotz (1983), the estimation of β_0 via maximum likelihood under the assumption that $E(\delta\delta') = \Sigma_0$ where Σ_0 is an arbitrary symmetric positive definite matrix, is

computationally burdensome if not infeasible with a large number of time periods. Thus, simplifying assumptions on the structure of Σ_0 will generally be necessary to implement ML. A number of these specifications are available in **HotzTran**, and are detailed in the next section. However, the consistency of estimators is not robust with respect to many of these simplifying assumptions. As an alternative, the estimation strategy we now outline has the advantage that it is computationally practical and possesses desirable large sample properties. This gain is achieved because our estimators require the evaluation of nothing more complex than period-by-period probit regression functions. Orthogonality condition (OC) estimators in multiperiod probit models can be motivated in the following way. Consider the conditional expectation of the observed zero/one y_{it}'s:

$$E[y_{it} | x_i] = P\{\delta_{it} > -x_{it}'\beta_0\} = F(x_{it}'\beta_0) \tag{4.1.1}$$

where $F(\ )$ is the standard normal distribution function. This implies the following regression equation:

$$y_{it} = F(x_{it}'\beta_0) + d_{it} \tag{4.1.2}$$

where d_{it} is a disturbance term orthogonal to all elements in x_i. For each individual we can specify a system of T such regressions and thus a vector of T disturbances. In particular, defining the residual vector:

$$H(y_i, x_i, \beta)' = [y_{i1} - F(x_{i1}'\beta), y_{i2} - F(x_{i2}'\beta), \ldots, y_{iT} - F(x_{iT}'\beta)], \tag{4.1.3}$$

with a typical element $H_t(y_{it}, x_{it}, \beta)$, then the "true" residuals are:

$$d_i' = (d_{i1}, d_{i2}, \ldots, d_{iT}) = H(y_i, x_i, \beta_0)'.$$

The form of these equations suggests nonlinear generalized least squares (GLS) as a natural estimation strategy for β_0. That is, β_0 can be estimated by choosing that β which minimizes:

$$\Sigma_{i=1}^{N} H(y_i, x_i, \beta)' \Omega_i^{-1} H(y_i, x_i, \beta) \tag{4.1.4}$$

where

$$\Omega_i = E(d_i d_i' | x_i). \tag{4.1.5}$$

Assuming, for pedagogical purposes, that Ω_i were known, the nonlinear GLS estimator of β, $\hat{\beta}_N$, would be the solution to the following first-order condition:

$$\Sigma_{i=1}^{N} \left[\frac{\partial H(y_i, x_i, \beta)}{\partial \beta} \Omega_i^{-1} H(y_i, x_i, \beta) \right] = 0 \tag{4.1.6}$$

here $\partial H(y_i, x_i, \beta)/\partial\beta$ is a $T \times k$ matrix with $-f(x_{it}'\beta)x_{it}'$ as a typical row, where $f(\ )$ is the standard normal density function. It can be shown that the GLS estimator is optimal among the class of estimators based on the fact that d_i is orthogonal to all the elements in x_i. In certain cases this estimator is asymptotically equivalent to ML. For example, if $T = 1$ or Σ_0 is

diagonal (thus, the δ_{it}'s from (2.1.4) are serially uncorrelated), it follows that Ω_i is also a diagonal matrix, with typical element ω_{it}. In this case (4.1.6) becomes:

$$\Sigma_{i=1}^{N}\Sigma_{t=1}^{T}\left[-f(x_{it}'\beta)x_{it}'\frac{H_t(y_{it}, x_{it}, \beta)}{\omega_{it}}\right] = 0 \tag{4.1.7}$$

and the GLS estimator is asymptotically equivalent to the ML estimator for the probit model with serially uncorrelated disturbances.

In general, the GLS estimator will not be as efficient as ML in cases where Σ_0 is not diagonal. This is because there is additional information contained in the orthogonality conditions formed with disturbances from regressions of y_{it} not only on x_{it} but the $y_{i\tau}$'s ($\tau \neq t$) and noncontemporaneous exogenous variables which are exploited in ML estimation but not GLS.

In practice, Ω_i is unlikely to be known since it depends upon not only Σ_0, but also x_i and β_0. A modified generalized least squares (MGLS) strategy is, therefore, of interest. Consistent estimators of Ω_i, utilizing consistent estimators of Σ_0 and β_0, could be used to obtain an estimator which is asymptotically equivalent to the GLS estimator. However, the estimators of the elements of Σ_0 are not easy to compute. The simplest strategy is to estimate each element of Σ_0 separately by maximizing the "conditional" likelihood function formed by bivariate probabilities of the y's for each pair of time periods, where one conditions this function on estimated values of β_0. Such estimators are consistent[3] and computationally feasible but are typically expensive to obtain. (They can, however, be computed in the program with the option TCORR = 1 (or 2 or 3).) Given the estimation of Σ_0, one still must construct Ω_i and invert it for each individual in the sample.

In many problems, implementation of the MGLS estimation strategy may be prohibitively expensive. Thus, we wish to find estimators of β_0 which do not necessitate estimation of the Ω_i's (and thus Σ_0). With this goal in mind, it is useful to view equation (4.1.6) in a different light. An inspection of the first-order conditions, (4.1.6), shows that the GLS (or MGLS) coefficient estimator is a special case of the class of estimators which are the form of linear weighting cross products of sample residuals and their derivatives with respect to β. That is, β is chosen to satisfy:

$$\frac{1}{N}\Sigma_{i=1}^{N} A_{Ni}G(y_i, x_i, \beta) = 0 \tag{4.1.8}$$

where G( ) is a $(T^2k) \times 1$ vector containing all possible cross products of the T residuals, $H_t(y_{it}, x_{it}, \beta)$, and the (T x k) matrix, $\partial H_t(y_{it}, x_{it}, \beta)/\partial\beta$, and A_{Ni} is a $k \times (T^2k)$ weighting matrix. In the case of GLS, A_{Ni} consists of zeros and elements of Ω^{-1}_i. Although this choice of A_{Ni} is asymptotically optimal within the class of estimators which satisfy (4.1.8) as we have argued previously, the coefficient estimator based on this choice may be difficult, in practice, to implement. For this reason, we consider estimators which constrain the A_{Ni}'s to be of the

---

[3]This assertion can be proved using an analogous argument to that used by Hansen (1982) in proving his Theorem 2.3. This method is discussed in Heckman (1978, page 949).

following form:

$$A_{Ni} = B_{1N}B_2(y_i, x_i, \beta_0), \tag{4.1.9}$$

where B_{1N} is a consistent estimator of a $k \times q$ constant matrix and B_2 is a $q \times (T^2k)$ matrix function of y_i, x_i and β_0 (or an estimate of β_0). Given a particular choice of B_2, we can index alternative estimators by choices of the weighting matrix B_{1N}. Furthermore, it is possible to choose and consistently estimate that B_{1N} which yields the smallest asymptotic covariance matrix among estimators which satisfy (4.1.8) and (4.1.9). The choice of B_2 may be guided by several considerations. On the one hand, one may choose B_2 so as to constrain the column dimensionality of B_{1N}. This feature is frequently important in insuring that estimation of an optimal B_{1N} is feasible. At the same time, B_2 may be chosen in order to allow for individual specific weighting of the elements of $G(\ )$ which depend on the unknown parameter β_0 but not Σ_0. Specification of the function B_2, a priori, enables us to estimate simultaneously these individual weighting matrices and β_0.

Thus we are led to consider estimators that solve minimization problems with first order conditions of the form:[4]

$$\tfrac{1}{N}B_{1N} \Sigma_{i=1}^{N} M(y_i, x_i, \beta), = 0 \tag{4.1.10}$$

where B_{1N} is a consistent estimator of the matrix B_1 and $M(y_i, x_i, \beta) = B_2(y_i, x_i, \beta)G(y_i, x_i, \beta)$. Such estimators which we denote as Orthogonality Condition (OC) estimators can be viewed as special cases of estimators studied by Hansen (1982). Hansen shows that for a broad class of weighting matrices, such estimators will be consistent and asymptotically normal. Of particular importance, he drives both the optimal choice of B_{1N} given $M(y_i, x_i, \beta)$, and expressions for the asymptotic covariance matrix of the estimators. This asymptotic covariance matrix accommodates, without modification, the conditional heteroskedasticity of d_i.[5] What makes this method particularly attractive is that computation of a consistent estimator of the optimal B_{1N} and of coefficient standard errors does not require a consistent estimator of Σ_0.

In **HotzTran**, we consider a specific choice of $B_2(y_i, x_i, \beta)$ (or equivalently of $M(y_i, x_i, \beta)$) for the multiperiod probit model. In particular, we consider the following weighted sums of residual and derivative cross products:

$$M_0(y_i, x_i, \beta) = \Sigma_{t=1}^{T} \left[\frac{-f(x_{it}'\beta)x_{it}}{[1-F(x_{it}'\beta)][F(x_{it}'\beta)]} H_t(y_{it}, x_{it}, \beta) \right] \tag{4.1.11}$$

---

[4]Alternative motivation and generalizations of this estimation strategy are considered in Avery, Hansen, and Hotz (1983).

[5]We note that the resulting asymptotic covariance matrix remains valid even if the off-diagonal elements of $E(\delta_i\delta_i'|x_i)$ are function of x_i.

$$M_{+j}(y_i, x_i, \beta) = \Sigma_{t=1}^{T-j} \left[\frac{-f(x_{it}'\beta)x_{it+j}}{[1-F(x_{it}'\beta)][F(x_{it}'\beta)]} H_t(y_{it}, x_{it}, \beta) \right] \quad (4.1.12)$$

$$M_{-j}(y_i, x_i, \beta) = \Sigma_{t=j+1}^{T} \left[\frac{-f(x_{it}'\beta)x_{it-j}}{[1-F(x_{it}'\beta)][F(x_{it}'\beta)]} H_t(y_{it}, x_{it}, \beta) \right] \quad (4.1.13)$$

for $j = 1, 2, \ldots, p$

where x_i^1 is a k_1 length subvector of x_i with elements that vary over time and where $p \leq T-1$ is the number of leads and lags actually used in estimation (set by option LAGS in the program). Thus we have grouped the cross product terms into three groups -- those involving residuals and contemporaneous x's, those involving residuals and leads x's, and those involving residuals and lagged x's. Note that each residual is weighted (for heteroskedasticity). We propose a class of OC estimators involving different B_1 matrices in (4.1.10) with M( ) defined to be:

$$M(y_i, x_i, \beta) = \begin{bmatrix} M_0(y_i, x_i, \beta) \\ M_{+1}(y_i, x_i, \beta) \\ \vdots \\ M_{+p}(y_i, x_i, \beta) \\ M_{-1}(y_i, x_i, \beta) \\ M_{-2}(y_i, x_i, \beta) \\ \vdots \\ M_{-p}(y_i, x_i, \beta) \end{bmatrix} \quad (4.1.14)$$

where the vector $M(y_i, x_i, \beta)$ has $q = k + k_1(2p)$ elements.

Before detailing our actual estimation procedure, let us make a few comments about the choice of $M(y_i, x_i, \beta)$ in (4.1.14). While our choice of the form of $M(y_i, x_i, \beta)$ (and thus of

the constraints (4.1.9)) is ad hoc and is intended to reduce computational burdens, the actual form of (4.1.11) through (4.1.13) is not completely arbitrary. To reduce the dimensionality of B_{1N}, we chose to treat the time periods symmetrically in the construction of $M(y_i, x_i, \beta)$. Second, we chose the individual specific weights so that $M_0(y_i, x_i, \beta)$ in (4.1.11) is the i^{th} individual component of the first derivative of the likelihood function when $E(\delta_i \delta_i' | x_i)$ is specified as the identity matrix. This latter feature implies that the ML estimator under the possibly incorrect assumption that the δ_i's are serially uncorrelated is a special case of our OC estimators.

Given $M(y_i, x_i, \beta)$, the choice of the weighting matrix B_{1N} in (4.1.10) is equivalent to the choice of W_N in the following criterion function:

$$C(\beta) = O_N(\beta)' W_N O_N(\beta) \tag{4.1.15}$$

where

$$O_N(\beta) = \frac{1}{N} \Sigma_{i=1}^{N} M(y_i, x_i, \beta),$$

are termed the "sample orthogonality conditions," and W_N is a consistent estimator of a $q \times q$ symmetric and positive semidefinite matrix W_0 with rank greater than or equal to k. A W_N matrix meeting this criteria will, in general, result in consistent and asymptotically normal estimators of β_0 for those β which minimize (4.1.15).

Alternative choices of W_N generate alternative OC estimators of β_0. Three choices are available in **HotzTran**. One estimator of interest is that which imposes only orthogonality conditions between residuals and contemporaneous x's. This estimator, b_N, is obtained by minimizing (4.1.15) when:

$$W_N = \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}$$

where I_k is a $k \times k$ identity matrix. It can be easily shown that the OC estimator is the quasi-ML estimator under the assumption that $\Sigma_0 = I$. Another available weighting matrix is:

$$W_N^* = \left[\frac{1}{N} \Sigma_{i=1}^{N} M(y_i, x_i, b_N) M(y_i, x_i, b_N)' \right]^{-1}.$$

The matrix W_N^* is a consistent estimator of $W_0^* = \{E[M(y_i, x_i, \beta_0)\ M(y_i, x_i, \beta_0)']\}^{-1}$ and produces an estimator of β_0 that has the smallest asymptotic covariance matrix among the estimators that minimize a criterion function of the form given in (4.1.15). It also has the advantage of being invariant to scale in the parameters and x's. W_N^* can can be formed from any consistent estimator, $\hat{\beta}_N$, not just b_N. We denote the estimator obtained using W_N^* in

criterion function (4.1.15) as b_N^*.

Finally, estimates can also be computed with the weighting matrix $W_N = I_q$ where I_q is a q x q identity matrix. This matrix effectively weights each orthogonality condition equally. It may, therefore, be sensitive to the scaling of the residuals and/or of the instruments. We denote these estimators as b_N^{**}.

Based on results in Hansen (1982), the asymptotic covariance matrix for an OC estimator, b_N, that minimizes (4.1.15) for a weighting matrix W_0 is given by:

$$(D_0'W_0D_0)^{-1}(D_0'W_0S_0W_0D_0)(D_0'W_0D_0)^{-1}, \tag{4.1.16}$$

where

$$D_0 = E\left[\frac{\partial M(y_i, x_i, \beta_0)}{\partial \beta}\right]$$

$$S_0 = E[M(y_i, x_i, \beta_0)M(y_i, x_i, \beta_0)'].$$

Consistent estimators of D_0, S_0, and W_0 are:

$$D_N = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{\partial M(y_i, x_i, \hat{\beta}_N)}{\partial \beta}\right],$$

where $\hat{\beta}_N$ is any consistent estimator of β_0 and

$$S_N = \frac{1}{N}\Sigma_{i=1}^{N} M(y_i, x_i, \hat{\beta}_N)M(y_i, x_i, \hat{\beta}_N)',$$

and W_N, respectively. Thus a consistent estimator of the asymptotic covariance matrix for b_N reduces to:

$$d_N^{-1}s_Nd_N^{-1}', \tag{4.1.17}$$

where

$$d_N = \frac{1}{N} \sum_{n=1}^{N} \left[\frac{\partial M_0(y_i, x_i, \hat{\beta}_N)}{\partial \beta} \right],$$

$$s_N = \frac{1}{N} \Sigma_{i=1}^{N} M_0(y_i, x_i, \hat{\beta}_N) M_0(y_i, x_i, \hat{\beta}_N)',$$

For b_N^* it reduces to:

$$(D_N W_N^* D_N')^{-1} \tag{4.1.18}$$

and for b_N^{**} it is:

$$D_N^{-1} S_N D_N^{-1}'. \tag{4.1.19}$$

It will always be the case that:

$$(D_N W_N^* D_N')^{-1} \leq d_N^{-1} s_N d_N^{-1}'$$

and

$$(D_N W_N^* D_N')^{-1} \leq D_N^{-1} S_N D_N^{-1}'.$$

Any of these estimated covariance matrices can be calculated with b_N, the computationally simpler of these estimators considered above. Thus estimated efficiency gains can be determined before deciding whether to calculate the asymptotically more efficient b_N^*.

Since b_N^* employs more orthogonality conditions than b_N and b_N^{**}, it may be of interest to test whether it is appropriate to impose the extra conditions. Such a test turns out to be a test of the exogeneity specification of the model. Other authors (e.g., Chamberlain (1982)) have noted that strict exogeneity imposes testable restrictions on the data and have proposed exogeneity tests. We propose an orthogonality condition test which is distinct but closely related to such tests. Using results in Hansen (1982), it can be verified that if $EM(y_i, x_i, \beta_0) = 0$, then

$$N \cdot O_N(b_N^*)' W_N^* O_N(b_N^*)$$

is asymptotically chi-square distributed with $2pk_1$ degrees of freedom. Thus one could test the strict exogeneity specification used in the model by comparing the value of the minimized criterion function using W_N^* with critical values for the chi-square distribution. This test is automatically computed and printed whenever the model utilizes over-identifying O.C.'s.

We can also utilize the fact that for any $\tilde{b}_N$ which minimizes (4.1.15).

$$N^{1/2} \Sigma_{i=1}^{N} M(y_i, x_i, \tilde{b}_N)$$

is asymptotically normal with asymptotic covariance matrix:[6]

$$T = [I_q - D_0(D_0'W_0D_0)^{-1}D_0'W_0]S_0[I_q - D_0(D_0'W_0D_0)^{-1}D_0'W_0]'. \quad (4.1.20)$$

T will be a singular matrix with rank q-k. Even though T is singular, Wald-type tests of the q-k over-identifying restrictions (or any subset of them) can be computed. Note that the restrictions need not actually be imposed in computation as a consistent estimate of T can be calculated from any consistent parameter vector. Thus, for example the inexpensive estimator, b_N, can be used to estimate D_N, S_N, and W_N^* -- hence T for the full set of orthogonality conditions. Thus, the "validity" of extra orthogonality conditions can be tested prior to imposing them. This procedure can also be used to test whether or not certain variables can be considered strictly exogenous. (See Avery, Hansen, and Hotz (1983) for examples of such tests.)

4.2. Implementing IV Procedures in HotzTran

The IV estimation procedures can be easily implemented in **HotzTran**. The simplest estimator, b_N, can be computed with the option TYPE = 1. These estimators are equivalent to ML under the i.i.d. assumption and only contemporaneous orthogonality conditions are imposed. The program will compute the correct standard errors as given in equation (4.1.17) if the option PSTAN = 1 is used. If PSTAN = 0 then standard errors will be computed under the assumption that errors are independent over time.

The more complicated IV estimators are computed with the option TYPE = 3. This option allows the separate specification of the expectation functions (residuals) and the instruments. If instruments from other time periods are used, the number of time leads/lags is determined by variable LAGS (equivalent to p in equations (4.1.12-4.1.13).). This can range from 0 (only contemporaneous O.C.'s in which case the estimates are equivalent to TYPE =1) to TIME - 1.

The orthogonality conditions are determined separately for each variable on the EQUATION VARIABLE CARD. It is possible to include a variable only in the computation of the residual, only as an instrument, only for forward orthogonality conditions (residuals orthogonal to future x's), only for backward orthogonality conditions (residuals orthogonal to lagged x's), only for contemporaneous orthogonality conditions (residuals orthogonal to contemporaneous x's), or any combination of conditions.

---

[6]This result is proved in Hansen (1982), Lemma 4.1.

Three types of weighting are allowed with TYPE = 3. O.C.'s can be weighted optimally by the option OPTWT = 0. This option will produce the estimator b_N^*. The coefficients used to compute the estimate of the optimal weighting matrix, $\hat{\beta}_N$, must generally be read in by the program (see START). O.C.'s can also be weighted equally with option OPTWT = 1. This option will produce the estimator b_N^{**}. In both cases the program will minimize the appropriate criteria function as in (4.1.15) (note though, that the actual printed function value is N times the function). When estimation is completed the program will printout any non-zero orthogonality conditions.

If regression, truncated, Tobit, or double Tobit models are estimated the program will add orthogonality conditions for the equation error variances (see the next section for details). This can be overridden by option SIGMA = 3 or 4.

A third weighting scheme is available in **HotzTran** with the option OPTWT = 2. This option provides a weighting scheme similar to that of traditional instrumental variables or two-stage least squares. It is applicable only if there are endogenous variables in the equation(s) or variables which are not included in the orthogonality conditions. The program will regress these variables against the entire set of instrumental variables specified, prior to program iteration. Coefficients from these "auxiliary" regressions will be used to compute fixed weights for the O.C's. With regression dependent variables, the estimated final coefficients will be identical to those of two-stage least squares. This option cannot be used with non-linear equations.

The program can compute standard errors in several ways if TYPE = 3. It will compute standard errors as given in equation (4.1.19) if the option PSTAN = 1 is used. With panel data S_0 is normally estimated by summing orthogonality conditions for all time periods of each cross-section then taking the crossproduct and adding up over all cross-sectional units. With the option MAVER these can be summed over a preset number of moving average lags or crossproducted for each time period (these options must be used with a pure time series). If PSTAN = 0 the standard errors will be computed under the assumption that the optimal weighting matrix is being used (see equation (4.1.18)).

The exogeneity test procedures described above can be easily implemented with the option TEXOG = 1. If used, a second set of orthogonality conditions (potentially different from those used in estimation) are read on the EQUATION TESTING VARIABLE CARD. The second set of orthogonality conditions must be at least as restrictive as those used in estimation. It is also possible to specify a longer lead/lag structure in the tested equations by TLAG. The program will automatically compute a chi-square test of the extra exogeneity conditions implied, and print out the value of each orthogonality condition and its standard error (note though, that because T in equation (4.1.20) is not of full rank some standard errors may be zero). If the EQUATION TESTING VARIABLE CARD is blank, the over-identifying orthogonality conditions of the model used in estimation will be tested. Note that the scale of the orthogonality conditions is not meaningful as it is sensitive to the scale of the x variables. Thus for any listing of the orthogonality conditions the ratio of their value to their standard error (t statistic in the printout) should be looked at.

Wald-type tests of any subsets of the over-identifying orthogonality conditions can be computed if TEXOG = 1. This is done by specifying TSUB > 0 and reading in a separate ORTHOGONALITY CONDITION TEST CARD for each Wald-type test.

Computation of the gains from utilizing the more expensive "optimally weighted" full or-

thogonality conditions estimators, b_N^*, can be calculated from the cheaper b_N estimators. b_N is first estimated using TYPE = 1. These estimates are then fed to the program as starting values and weighting matrix values (START = 0) for TYPE = 3. Any desired full O.C.'s are specified on the EQUATION VARIABLE CARD. The optimal weighting matrix is specified (OPTWT = 0). The optimally weighted estimates need not actually be computed though. If the ESTIMA = 4 option is used, the program will compute an estimate of the standard errors of the optimally weighted estimators without actually computing the parameter estimates. Potential efficiency gains can be measured relative to the standard errors of the b_N estimators.

If ESTIMA = 4 it is also possible to test the extra orthogonality conditions implied in the b_N^* estimators utilizing b_N. Setting TYPE = 3, OPTWT = 0, and again reading in b_N, the EQUATION VARIABLE CARD should specify only contemporaneous orthogonality conditions. The full orthogonality conditions should be specified on the EQUATION TESTING VARIABLE CARD with TEXOG = 1. The over-identifying restrictions will be automatically tested with subsets testable by setting TSUB > 0. Note that the over-identifying restrictions cannot be simultaneously tested with the computation of the prospective standard error gain just mentioned, since each ESTIMA = 4 computation requires a different specification of the O.C.'s on the EQUATION VARIABLE CARD.

4.3. Implementing other Models with IV Estimation

The (IV) estimation procedures described in the previous two sections can be extended to non-probit models and to multiple-equation problems. The application of the procedures to other model forms such as Tobit, double Tobit, regression, logit and truncated regression is straightforward. The principle is identical to the probit model -- weighted cross-products of residuals and contemporaneous, lead, and lagged x's are set to zero. Like the probit model the residuals used are weighted such that the contemporaneous orthogonality conditions will be identical to the first derivatives of the i.i.d. log-likelihood function. We do not present these for the logit and truncated models as they can be looked up.

For Tobit and double Tobit masspoint observations, the orthogonality conditions comparable to (4.1.11) are:

$$M_{0t}(y_{it}, x_{it}, \beta, \sigma) = \left[\frac{-f(x_{it}'\beta/\sigma)x_{it}/\sigma}{[1-F(x_{it}'\beta/\sigma)][F(x_{it}'\beta/\sigma)]} H_t(y_{it}, x_{it}, \beta, \sigma) \right] \tag{4.3.1}$$

where,

$$H_t(y_{it}, x_{it}, \beta, \sigma) = 1 - F(x_{it}'\beta/\sigma)$$

and,

$$H_t(y_{it}, x_{it}, \beta, \sigma) = -F(x_{it}'\beta/\sigma)$$

for upper and lower masspoints, respectively. Note that the equation error variance, σ, has been added as an estimated parameter. For regression, and the Tobit, and double Tobit non-masspoint observations, the orthogonality conditions comparable to (4.1.11) are:

$$M_{0t}(y_{it}, x_{it}, \beta, \sigma) = ((y_{it}-x_{it}'\beta)/\sigma)x_{it}/\sigma \tag{4.3.2}$$

To obtain the full orthogonality conditions, the M_t's are summed over observations. Thus for Tobits the orthogonality conditions will be a mixture of masspoints and non-masspoints.

As with the probit model, orthogonality conditions can be created by crossing residuals with lead and lagged x's if panel data are used. When estimating Tobit and regression models, it is also possible to add an orthogonality condition from the first derivative of σ in the i.i.d. log-likelihood function. This orthogonality condition, which is summed only for the non-masspoint observations, is:

$$(((y_{it} - x_{it}'\beta)/\sigma)^2 - 1)/\sigma \tag{4.3.3}$$

Under normal conditions the σ orthogonality condition will always be included whenever TYPE = 3 is used. In general it will not be possible to identify σ unless this orthogonality condition is included (or restrictions imposed). However, particularly with regression models, this may cause estimation problems. The β and σ orthogonality conditions can generally be driven to zero by driving σ to infinity as well as being satisfied at their "correct" values. Thus particularly with bad starting values, the estimation procedure may blow up. There is a good solution to this problem with regression models. The σ orthogonality condition can be dropped by setting SIGMA = 3,4 and by use of the RESTRICTION CARD. σ can be fixed at an appropriate value and not estimated. Because of the separation of β and σ in a regression model, the estimates of β will be consistent. When β has been estimated, σ can be "unfrozen" and estimated as well. Unfortunately, the same procedure cannot be as easily used for Tobit models, although the same problem is present, as estimation of β and σ is not separable. Thus σ must be fixed at a consistent value in order for estimates of β to be consistent. These could be obtained from an initial run with TYPE = 1 or from alternative sources. If this is not possible, σ should not be fixed, and with good starting values it hopefully will not converge to infinity. These problems should occur only with TYPE = 3.

One interesting feature available with regression models is the relaxation of the homoskedastic error assumption with panel data, even in the case where the form of the heteroskedasticity is unknown. Models of this type can be consistently estimated utilizing the same principle that underlies estimation of multiple-time period or equation models even when the form of their error covariance is unknown. If σ is fixed at an arbitrary value, the program will consistently estimate β and will printout correct asymptotic standard errors when TYPE =

1 and PSTAN = 1 or TYPE = 3 are used with panel data.

The principles of O.C. or IV estimation can also be applied to the multiple-equation setting (see Avery and Hotz (1984)). The essential principle underlying these methods is to form orthogonality conditions between the residuals imbedded in (4.1.11), (4.3.1), and (4.3.2) and auxiliary x's which come from other equations or sources. This principle is what underlies seemingly unrelated regression, 2SLS, and instrumental variable regression. In **HotzTran** the use of the EQUATION VARIABLE CARD allows the specification of such instrumental variables. Instruments can be used which are not included in the equation. It is also possible to include variables in the formation of the equation, but not to set them orthogonal to the residual (or only use lead or lagged values in the orthogonality conditions).

This feature is particularly useful for estimating simultaneous equations models and for Rational Expectation Type specifications. The user has the option, in such cases, of either using the set of instruments to estimate the model of interest in one pass, or to construct predicted values for each endogenous variable and then use those predicted values, along with the other exogenous variables as instruments in the second stage regression of the "structural" equation or equations. This latter process is simply two-stage least squares. **Note:** The 2SLS procedure can be implemented in this program, but the standard errors of estimates will not generally be correct. Finally, as illustrated in the discussion of OC estimation above, one can utilize instruments from other time periods in the estimation of current period equations.

4.4. Sample Selection Models

HotzTran can also be used to estimate equations or systems of equations which are subject to a stochastic sample inclusion criterion. Suppose that the *behavioral equation* one is interested in estimated is given by a latent equation like (2.1.4) with an observed probit, Tobit, double Tobit, or regression indicator. Now suppose that the *sample selection decision* is governed by a probit equation in which the the latent index is of the form:

$$y^*_{Sit} = z'_{it}\gamma_0 + \epsilon_{it} \tag{4.4.1}$$

where the i^{th} observation in the t^{th} time period is included in the sample iff $y^*_{Sit} > 0$. Note that it is possible that z_{it} may share some variables in common with x_{it} in equation (2.1.4) and there may be restrictions across the elements of γ_0 and β_0. Finally, for the sample selection criterion to affect the estimation of equation (2.1.4), the disturbances δ_{it} and ϵ_{it} must be correlated.

HotzTran is capable of generating ML estimates of this type of sample selection model. The program does this by viewing the behavioral equation and equation (4.4.1) as a two equation system which has correlated errors. The sample inclusion equation is estimated as a probit. The behavioral equation is assumed to be observed if and only if the probit indicator of the sample inclusion equation is one. The resulting estimators of γ_0 and β_0 are consistent and the program calculates an estimate of their asymptotic standard errors. To implement the ML procedure, set TYPE = 4 and use the UOBSV option on the BASIC EQUATION COMMAND to include the behavioral equation only when the dependent variable of the sample selection equation is one. If there is more than one behavioral equation, they can all be es-

timated jointly using the TYPE = 4 procedure although estimates will generally only be "quasi-ML".

We note that one could also estimate this type of sample selection model using a a two-step procedure suggested by Heckman (1978). To implement this, first estimate the sample inclusion equation separately as a probit using TYPE = 1. Using the option RWRITE, write out the residuals on disk. The fourth item of each residual line can then be included as an independent variable in the behavioral equation(s). This will properly correct for sample selection bias in those equations. Note though that the program will not generate the correct standard errors in the second stage of this procedure. This procedure can be used to estimate the types of selection bias models in panel data contexts discussed by Griliches, Hall and Hausman (1978) and Maddala (1978). An example of the implementation of the two-stage procedure is given in Chapter 5.

4.5. MIMIC Models with Limited Dependent Variables

The specification of the MIMIC model with discrete (or more generally, limited dependent) and continuous variables can be described as follows. A p x 1 vector of continuous indicators, y^*, is assumed to be a linear function of an m x 1 vector of common latent variables, η, and a vector of specific factors, ϵ. That is

$$y^* = \alpha + \Lambda\eta + \epsilon \tag{4.5.1}$$

where α (p x 1) and Λ (p x m) are fixed parameters and $p \geq m$. It is assumed that η is related to a q x 1 vector of observable causal variables, x, by:

$$B\eta = \Gamma x + \zeta \tag{4.5.2}$$

where B (m x m) and Γ (m x q) are fixed parameters and ζ is a random vector. Both ϵ and ζ are assumed to be normally distributed with zero means and covariances T (p x p) and Ψ (m x m), respectively. In keeping with the view that the ϵ's are specific factors it is assumed that T is a diagonal matrix. Finally, it is assumed that x contains only strictly exogenous variables, i.e., $E(\epsilon \mid x) = 0$ and $E(\zeta \mid x) = 0$, and that all variables are independent and identically distributed across observations. Note that in the most general setting the x's can be considered to be measured with error. We do not consider that case here.

The reduced form relationship between y^* and x can be obtained by solving (4.5.1) and (4.5.2):

$$y^* = \alpha + \Pi x + u \tag{4.5.3}$$

where $\Pi = \Lambda B^{-1}\Gamma$, and $u = \epsilon + \Lambda B^{-1}\zeta$ is the reduced form disturbance vector with covariance matrix:

$$E(uu' \mid x) = \Sigma = \Lambda B^{-1}\Psi B^{-1\prime}\Lambda' + T. \tag{4.5.4}$$

The structure of the model is in terms of the restrictions on Π and Σ. As can be seen in equations (4.5.3) and (4.5.4), a MIMIC model implies restrictions both within and across rows of Π and Σ and restrictions across the elements of Π and Σ. Note that if $\Pi = 0$, the specification reduces to a factor model.

In the traditional MIMIC model described by Jöreskog and Goldberger (1975), elements of y^* are assumed to be continuous and observable. While maintaining the continuity assumption, we wish to consider cases where we have limited observations of some of the elements of y^*. We will assume that what is observed is a vector y (p x 1) whose elements are of the form of probit, Tobit, double-Tobit, or regression models.[7]

In **HotzTran**, OC or IV estimation is used to estimate structural parameters of MIMIC models with limited dependent variables. A more thorough discussion of the implementation of this method to such models is found in Avery and Hotz (1982). Here we briefly show how **HotzTran** goes about estimating the the unique elements of α, Λ, B, Γ, T, and Ψ.

Three methods can be used in **HotzTran** to estimate MIMIC-type models. Two of the three methods utilize **HotzTran** to estimate the reduced form (4.5.3). If the elements of the covariance matrix, Σ, are not needed to identify structural parameters, the multiple-equation system (4.5.3) can be estimated with TYPE = 1. Alternatively, if Σ elements are needed for identification, (4.5.3) can be estimated with TYPE = 4. Both sets of estimates will be "quasi-ML" (except TYPE = 4 with two equations). In either case, each equation is specified separately and can have a probit, Tobit, double Tobit, or continuous indicator of y^*. With very simple MIMIC structures, the reduced form (4.5.3) can be estimated unrestricted. Alternatively, restrictions can be imposed (see RESTRI) across equations. The restrictions implied by MIMIC models are generally proportional, and can be imposed easily. With more complicated structures, it may be necessary to use the NONLIN option and spell out the models' structure (the error variances and covariances can also be specified when TYPE = 4 is used). Unfortunately, this will require that the user solve the system for the reduced form representation of the system in terms of the structural parameters. In practice this may make this an undesirable option in some cases.

The third method is much simpler, but requires the availability of another program such as LISREL. Each dependent and independent variable (including probit or Tobit indicators) is fed into the system. TYPE = 4 is used with a separate equation specified for each variable. Each "equation" includes only a constant as a regressor (thus use blank EQUATION VARIABLE CARD if CONST = 0). Using regression starting values (START = 2) should yield rapid convergence. The option MWRITE should be used. This will write out (punch) and estimate the

---

[7]Such MIMIC models, in this more general limited dependent variable case, are discussed in Maddala (1980). The models he describes can be estimated using **HotzTran**.

unconditional covariance matrix of the y^*'s (not the observed y's) and the x's. The estimated unconditional covariance matrix can be input to a program like LISREL, restrictions imposed, and the structural parameters solved for, just as if all y^*'s (including probits/Tobits) were observed. In general, this two step procedure will yield consistent estimates of the structural parameters. For more on this latter approach see Avery and Hotz (1982).

4.6. Estimation of Nonlinear Regression Models

While already discussed previously, in this section we offer a few more comments on the convenient features within **HotzTran** to estimate nonlinear models. As mentioned previously in the discussion of regression models, the program is capable of estimating nonlinear equations or systems of equations subject to the restriction that equation errors are additive. Such equations may be nonlinear only in the parameters--due to proportional cross-equation restrictions, for example--or they may be nonlinear in both the parameters and the variables. Models which have nonlinearities only in the parameters can be estimated fairly inexpensively using the RESTRI option. This option allows cheaper methods employed in standard linear estimation -- such as using moment matrix sufficient calculations whenever possible -- to be used. More complex nonlinear problems require the NONLIN option which can be more expensive. This option requires that the program sum observations for each iteration and uses no moment information.

Either nonlinear capability -- RESTRI or NONLIN -- can be combined with other options. For example they can be used with any model form (such as probit etc.) or with any estimation procedure (i.e. ML or IV). The program allows either the restrictions or the nonlinear equations to be written in a "user friendly" form. The user need only write out the restriction or equation and

the program will automatically (and correctly) determine the gradients (first derivatives of the objective function) needed to perform iterative search in parameter estimation. Second derivatives used in calculation of parameter standard errors will also be correctly derived. There are certain constraints on the actual form of the nonlinear equations allowed and in the types of parameter restrictions which can be imposed. These are described in the discussion of the NON-LINEAR EQUATION CARD(S) and the RESTRICTION CARD(S).

Given that estimation of arbitrary nonlinear equations is often a difficult task, we offer a few suggestions to the user concerning the estimation of specifications.

- For complicated equations, the objective function (the likelihood function or the IV objective function) may not be globally concave in the parameters; so local optima can occur. Thus to insure that the global optimum of the objective function has been found, one should perturbate "converged" parameters to insure that there are not parameter values which significantly improve the likelihood function or the IV criterion function.

- Secondly, when using IV estimation, the user should be aware that, except when using those orthogonality conditions which have been optimally weighted, the scale of these conditions can affect the optimization procedure and the rate of convergence. For example, since one generally does not have enough enough information to form optimal weights when one is first beginning to fit a model, one is generally forced to start with an identity matrix as the weighting matrix. But using the identity

matrix will generally mean that the rate of convergence will be very slow or the iterative search for the optimum may actually "stall." That is, the minimizing algorithm may have difficulty finding a way to significantly improve the objective function. Such situations can be particularly problematical because the iterative algorithms may move into "bad" regions of the parameter space (i.e., inappropriate values of the parameters being chosen) and may have difficulty getting out of such regions in future iterations. In such situations, it is sometimes helpful to use the intermediate parameter estimates to form an "optimal" weighting matrix and to use that weighting matrix in the estimation of those parameters even though such intermediate estimators are not consistent. Such a scheme may help one to weight the orthogonality conditions properly and make them less subject to the scale of the orthogonality conditions being used. Note also, that when using the IV estimation technique (TYPE = 3), one must generally constrain the σ to some value during the first stage of estimation. The σ parameter can be freed and estimated after the other parameters have been estimated. This procedure should be followed only with regression models. It is not necessary with TYPE = 1.

- Finally, every effort should be made to get "good" starting values. The closer the user is able to come to starting parameters in the neighborhood of the values which optimize the objective function, the fewer troubles are likely to occur. For complicated models with cross-equation restrictions, the "unrestricted" versions of the model can be run to get some idea as to what ranges the "structural" parameters might take. Also one might try a sequential strategy of estimating simpler, nested specifications (e.g., restricting some of the coefficients to be zero or to take on certain fixed values) and then use the estimates from such runs to start of the expanded model. It may also be useful to use only a subset of the observations (preferably random) when experimenting.

5. EXAMPLES OF SAMPLE RUNS

The following are a series of sets of CARDs to run various equation types and systems of equations. To facilitate the presentation of these examples, we shall first show how one would read in a basic data set, then present a number of examples of manipulations that might be done and models that might be estimated with this data.

5.1. Reading Data Into HotzTran

Suppose we have women's work data on 120 female-headed families for five years each, thus have a total of 600 observations. These could be used in **HotzTran** as 600 separate observations or in a way that takes explicit account of the panel nature of the data. To do the latter it is necessary to presort the data. The observations must be sorted first by family, then within each family, by time period. Thus the first observation would be family 1 time period 1. The next would be family 1 time period 2 etc. To complete our example, suppose we have the following ten variables for each of the families:

LFP	A dummy variable equal to 1 if the woman worked during the year and 0 otherwise
HOURS	The total number of hours the woman worked in each year
WAGE	The wage rate the woman received in each year in $ per hour
FERT	A dummy variable equal to 1 if the woman gave birth to a child during the year and 0 otherwise
EDUC	Number of years of education for the woman (constant over time)
INCOME	Family income in each year
AGE	The woman's age in each year
KIDS	The number of children the woman had given birth to as of the beginning of the year
HUS	A dummy variable equal to 1 if the woman had a husband and 0 otherwise (constant over time)
RACE	A dummy variable equal to 1 if the family was white (constant over time)

Thus, for example, the first 10 lines in the data set, which consist of the data for the first individual for each of the five years and the data for the second individual for each of the five years, might look like the following:

```
1.0   500.  4.00  0.0   12.0  12000. 25.0 0.0   1.0   1.0
0.0   000.  0.00  0.0   12.0  12500. 26.0 0.0   1.0   1.0
1.0   2100. 4.50  0.0   12.0  13500. 27.0 0.0   1.0   1.0
0.0   000.  0.00  1.0   12.0  22000. 28.0 1.0   1.0   1.0
0.0   000.  0.00  0.0   12.0  20000. 29.0 1.0   1.0   1.0
0.0   000.  0.00  0.0   11.0  18000. 37.0 4.0   1.0   1.0
0.0   000.  0.00  0.0   11.0  20000. 38.0 4.0   1.0   1.0
0.0   000.  0.00  0.0   11.0  11500. 39.0 4.0   1.0   1.0
1.0   330.  3.10  0.0   11.0  12000. 40.0 4.0   1.0   1.0
1.0   1200. 3.90  0.0   11.0  18000. 41.0 4.0   1.0   1.0
                          .
                          .
                          .
```

where each line has a format of (10F6.0).

The following are descriptions of the DATA CONTROL CARDs, RUN TITLE CARDs, VARIABLE NAME CARDs, and DATA TRANSFORMATION CARDs (if used) needed to read in the data. Suppose first that the data is on a disk file designated as device 17 in the JCL in format 10F6.0, and we want to read in the data and print the means using normal 132 character print. The following control cards would be used (note that all commands set at their default value need not be specified):

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10 PRINT=1 LINE=0
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
```

Suppose that we want to save the data for future runs and want 80 character printing. Then:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10 PRINT=1 $
DISK=2 LINE=1
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
```

This will write the observations out in single precision binary format on device INT (set to 18). In this example, device 18 should be set with LRECL=40 when using IBM MVS/TSO.

To read data in for subsequent runs from this device (cheapest runs) and bypassing the print, use:

```
TIME=5 OBSV=120 VARI=10 INPUT=18 DISK=1
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
```

This will bypass the data read section and is the cheapest to run. To perform some transformations enter:

```
TIME=5 OBSV=120 VARI=10 INPUT=17 FORMAT=(10F6.0) TRANS=4
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
V11=[HOURS]**2
V12=[WAGE]**2
IF([RACE].EQ.1) V13=[INCOME]
V14=VT5
HOURSSQ   WAGESQ RACE*INC YEAR5
```

Note that DISK = 1 can not be used with transformations.

Often data such as this can have "holes". Suppose, for example, that if there is no husband present, that data for fertility and kids is missing and coded as -99. and that complete data is not available for all time periods for all families. In particular suppose that a total of 123 observations are missing from the potential 600 and that variable V11 contains the value (1-5) of the time period for each observation. Then:

```
OBSV=477,TIME=V11,TIME=5,INPUT=17,FORMAT=(11F6.0),VARI=11,MISS=-99,$
    PRINT=1
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TIMEPERI
```

If you were unsure how many time periods were missing. you could set OBSV=600 and have the program read to end of file.

Following up on this example, suppose that you wanted to do all of the above but were using SUBROUTINE USER and that variable V11 had to be converted from years (1980-1984) to time periods (1-5), and that the only way that you could tell that an observation was for a new family was if a variable V12 = 1. Then:

```
//FORT.SYSIN   DD *
      SUBROUTINE USER(X,NVAR,INUM,INN,IDD)
      READ(INN,END=100,10) (X(I),I=1,11),X12
   10 FORMAT(12F6.0)
      X(11)=X(11)-1979
      IF(X12.EQ.1) X(11)=-X(11)
      RETURN
  100 IDD=3
      RETURN
      END
//GO.FT17F001 DD   DSNAME=DATAFILE.DATA,UNIT=SYSDA,DISP=SHR,
     DCB=(RECFM=FB,LRECL=48,BLKSIZE=480)
//GO.SYSIN  DD  *
OBSV=600 TIME=V11 TIME=5 FORMAT=USER MISS=-99 VARI=11 INPUT=17 $
    PRINT=1
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TIMEPERI
```

The lines starting with // are IBM JCL and wouldn't be used on other systems. If just checking data, each of the above examples could be run followed by a line with STOP on it.

In the next section we present a number of simple example runs that could executed using this data.

5.2. Simple Run Examples

To run a simple probit model on the woman's likelihood of working given her fertility, education, age, kids, and race, run:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=1 START=2 ITERA=20
DEP=[LFP].GT.0 MODEL=PROBIT
FERT EDUC AGE KIDS RACE
STOP
```

Note that we have listed the model's independent variables in numeric order (as they appear in the data file). However, the listed order is irrelevant, as the program will always reorder coefficients by the numeric order of the variables, followed by the constant.

To run a similar Tobit model on the woman's hours worked, the last three lines would be:

```
DEP=V2.GT.0 MODEL=TOBIT
V4 V5 V7 V8 V10
STOP
```

These examples would use standard convergence criteria with up to 20 iterations allowed. Starting values for each model would be obtained from regressions run by the program. A constant would be used in each model.

These model runs take no account of the panel nature of the data. Note that EDUC and RACE are constant over time. Some storage space could be saved using the TIME VARIATION CARD, therefore. Suppose as well that we want to compute standard errors to take the panel data nature properly into account. Then the model run cards might be:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=1 ITERA=20 START=2 PSTAN=1
EDUC=C RACE=C
MODEL=TOBIT DEP=[HOURS]
V4 V5 V7 V8 V10
STOP
```

There are other ways to take more explicit account of the inter-temporal correlations of the equation errors. The following run would execute first a random effects model, then a model with a first order autoregressive error, and finally, a model with a fixed effect.

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=2 RANDOM=0 HERMITE=9 START=2 ITERA=20
DEP=V2 MODEL=TOBIT
V4 V5 V7 V8 V10
TYPE=2 RANDOM=1 HERMITE=9 START=2 ITERA=20
DEP=V2 MODEL=TOBIT
V4 V5 V7 V8 V10
TYPE=1 START=2 ITERA=20 PSTAN=1
DEP=V2 MODEL=TOBIT FIXED=1
V7 V8 V4
STOP
```

Note that since the fixed effect model is equivalent to putting a separate constant term in the model for each family, the variables EDUC and RACE which are constant over time must be dropped. Note as well that there will be no explicit constant in this run.

To run the model without taking account of inter-temporal errors, and then compute an estimate of the correlation of errors from the five time periods, you might use the following run cards:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=1 ITERA=20 START=2 TCORR=2 PSTAN=1
DEP=V2 MODEL=TOBIT
V4 V5 V7 V8 V10
STOP
```

5.3. Multiple Equations, Observation Subsets, and Non-Linear Models

The previous examples are all straightforward runs with few complications except the panel data. The following examples are somewhat more complicated. We start with runs which use a subset of the data. Suppose we only wanted to use data where the husband related data was not missing, and only wanted to use the first four time periods, and only white families. The run might look like this:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10 MISS=-99
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=1 START=2 ITERA=20 PSTAN=1 UTIME=1TO4 UOBSV=[RACE].EQ.1 MISS=1
DEP=V2 MODEL=TOBIT
V4 V5 V7 V8 V10
STOP
```

Each of the previous models had only a single equation. Suppose that you want to estimate the wife's fertility and work decisions jointly. The following run would estimate such models two ways -- first without estimating an equation correlation, and second with maximum likelihood estimates of the two equations and their error covariance.

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=1 EQUA=2 ITERA=20 PSTAN=1 START=2
DEP=V2 MODEL=TOBIT
V5 V7 V8 V10
DEP=V4 MODEL=PROBIT
V5 V7 V8 V10
TYPE=4 EQUA=2 ITERA=20 PSTAN=1 START=2 RHO=1
DEP=V2 MODEL=TOBIT
V5 V7 V8 V10
DEP=V4 MODEL=PROBIT
V5 V7 V8 V10
STOP
```

We modify this run in two ways. First the cheaper joint equation estimates without error correlation estimates (TYPE = 1) can be modified to compute "quasi-ML" estimates of the error correlations after coefficients are estimated. Secondly, we modify the TYPE = 4 estimates to impose the restriction that the variables have a proportional effect in the two equations.

```
TYPE=1 EQUA=2 ITERA=20 PSTAN=1 START=2 ECORR=2
DEP=V2 MODEL=TOBIT
V5 V7 V8 V10
DEP=V4 MODEL=PROBIT
V5 V7 V8 V10
TYPE=4 EQUA=2 ITERA=20 PSTAN=1 START=2 RHO=1 RESTRI=3
DEP=V2 MODEL=TOBIT
V5 V7 V8 V10
DEP=V4 MODEL=PROBIT
V5 V7 V8 V10
B8=B2*B7/B1
B9=B3*B7/B1
B10=B4*B7/B1
STOP
```

Note, that in counting coefficient order, the Tobit equation error sigma follows the four variable coefficients and constant for the first equation. The five coefficients of the probit equation follow (constant last).

Non-linear equations can be introduced by the following two-model run. The first model is identical to the restricted model above (although note that coefficient order is different since the Tobit sigma now comes after all variable coefficients). The second model depicts similar equations in a multiplicative model with an additive normal error.

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=4 EQUA=2 ITERA=20 PSTAN=1 START=0 NONLIN=2
DEP=V2 MODEL=TOBIT
DEP=V4 MODEL=PROBIT
E1=B1*V5+B2*V7+B3*V8+B4*V10+B5
E2=B6*V5+B2*B6*V7/B1+(B3*B6/B1)*V8+V10*B4*B6/B1+B7
(9F4.0)
 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0  .5
TYPE=4 EQUA=2 ITERA=20 PSTAN=1 START=0 NONLIN=1
DEP=V2 MODEL=TOBIT
DEP=V4 MODEL=PROBIT
E1=V5**B1*V7**B2*V8**B3*B4
S1=B9
E2=V5**B5*V7**B6*V8**B7*B8
C=B10
(10F4.0)
 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0  .5
STOP
```

Note, that since we are using non-linear models, coefficient starting values must be read in. Note as well, that although we used the full non-linear system (NONLIN=1) in the second model, the cheaper version (NONLIN=2) could have been used since the error sigma and covariance terms (S1 and C) are constants.

5.4. Sample Selection Bias Models

In Chapter 4 we presented a brief discussion of sample selection bias models, where one equation "censors" the appearance of one or more other equations. Suppose, for example, that we wanted to estimate an equation for a woman's wage; however, the wage will be observed only if she works. Thus the labor force participation decision "censors" the wage equation. Since, the labor force equation errors are likely to be correlated with the errors of the wage equation we cannot use TYPE = 1. TYPE = 4 can be run to provide ML estimates of a censoring HOURS worked labor force equation and a wage equation as follows.

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       JOINT EQUATIONS FOR HOURS AND WAGES
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=4 EQUA=2 ITERA=20 START=2 PSTAN=1
DEP=V2 MODEL=TOBIT
V5 V7 V8 V10
DEP=WAGE MODEL=REGRES UOBSV=V2.GT.0
V5 V7 V8 V10
STOP
```

Note that the UOBSV command for the WAGE equation means that it will only use observations when women work (HOURS > 0). Because regression starting values will not be consistent for the sample selection bias models, TYPE = 4 runs may occasionally have trouble converging or be unable to give standard errors (particularly for the equation correlation). It may also be the case that many observations are being used, and a number of equations explored,

which might make TYPE = 4 runs expensive. An alternative strategy, therefore is to use the procedure advocated by J. Heckman. This procedure, which takes two steps, can be implemented in **HotzTran** as follows. First run the Tobit (a probit on work/no work would be equally applicable) on HOURS by itself (there is no censoring of this equation), saving the residuals on file 8 (SAVE.DATA) by the option RWRITE.

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       JOINT EQUATIONS FOR HOURS AND WAGES
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=4 START=2 RWRITE=8
DEP=V2 MODEL=TOBIT
V5 V7 V8 V10
STOP
```

Heckman's procedure requires saving the expectation of the HOURS equation error term conditional on the woman working. If the censoring equation is a Tobit, a Probit, or a Double Tobit, this term is stored as the fourth term of the RWRITE file. In the second stage this term is added as a regressor in the WAGE equation. It can be added easily to the rest of the data set by using SUBROUTINE USER.

```
//FORT.SYSIN   DD *
      SUBROUTINE USER(X,NVAR,INUM,INN,IDD)
      READ(INN,10) (X(I),I=1,10)
   10 FORMAT(10F6.0)
      READ(8,20) X(11)
   20 FORMAT(48X,D16.9)
      RETURN
      END
//GO.FT08F001 DD   DSNAME=SAVE.DATA,UNIT=SYSDA,DISP=SHR,
     DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//GO.SYSIN  DD  *
OBSV=120 TIME=5 INPUT=17 FORMAT=USER VARI=11
       JOINT EQUATIONS FOR HOURS AND WAGES
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE HOURERRO
TYPE=1 START=2 PSTAN=1
DEP=V3 UOBSV=V2.GT.0 MODEL=REGRES
V5 V7 V8 V10 V11
STOP
```

The lines starting with // are IBM JCL and wouldn't be used on other systems. The standard errors in **HotzTran** for the two-stage Heckman procedure will be biased, as they do not take the estimation of the first stage into account. Coefficients will be consistent.

Finally suppose that we want to estimate fertility, work hours, and wage equations together, looking at women with husbands present. Although coefficients will not be ML any more, **HotzTran** can handle this three equation system with sample selection bias easily with TYPE = 4.

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=4 EQUA=3 ITERA=20 PSTAN=1 START=2
DEP=V2 MODEL=TOBIT
V5 V7 V8 V10
DEP=V4 MODEL=PROBIT
V5 V7 V8 V10
DEP=WAGE MODEL=REGRES UOBSV=V2.GT.0
V5 V7 V8 V10
STOP
```

5.5. Running Instrumental Variable Models

In this section we present several simple examples of the use of instrumental variable (or orthogonality condition) estimation. Let us begin with a relatively simple model of women's work hours, where her current work hours are a function of age, education, race, and other family income. However, suppose that we also want her current work hour equation errors to be orthogonal to lagged family income (although it is not explicitly in the equation). This equation could be estimated by the following (Note that we have changed the convergence criteria to gradients only as this seems to work better with TYPE = 3):

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=3 ITERA=20 CNVRG=2 PSTAN=1 START=2 LAGS=1 OPTWT=1
DEP=V2 MODEL=TOBIT
V5 V6=3 V7 V10
STOP
```

This model uses equal weighting of the orthogonality conditions since we lack consistent values of the parameter matrix with which to estimate the optimal weights. This could be changed by running TYPE=1 first to obtain consistent parameter estimates (even though the lagged instrument is left out). We could also test the model's extra orthogonality condition by adding a blank orthogonality testing card.

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=3 ITERA=20 CNVRG=2 PSTAN=1 START=4 LAGS=1 OPTWT=0 TEXOG=1
DEP=V2 MODEL=TOBIT
V5 V6=3 V7 V10

STOP
```

If we wanted to test all the lags of income jointly and individually this would be accomplished by the following:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=3 ITERA=20 CNVRG=2 PSTAN=1 START=4 LAGS=1 OPTWT=0 TEXOG=1 $
  TSUB=4 TLAG=4
DEP=V2 MODEL=TOBIT
V5 V6=3 V7 V10
V6=3
SUBSET=3
SUBSET=4
SUBSET=5
SUBSET=6
STOP
```

Finally, the instrumental variable features of **HotzTran** can be used to estimate rational expectations forecasting equations such as implicit function first order conditions which have appeared recently in the literature. Although the following is somewhat contrived, it is an example of what could be done. Suppose that theory tells us that the difference between a woman's actual hours worked, HOURS, and a forecast of hours based on lagged exogenous variables, is independent of all variables in existence the previous period. To estimate such a model we can only use four time periods and must create a new variable, HOURS+1, which is future hours. We estimate the equation as a regression (ignoring the Tobit) and as a nonlinear (concocted) implicit equation:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10 TRANS=2
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
V11=[HOURS](+1)
V12=[AGE]-[EDUC]+6
HOURS+1 EXPER
TYPE=3 ITERA=20 CNVRG=2 PSTAN=1 START=0 OPTWT=1 NONLIN=2 UTIME=1TO4 $
    RESTRI=1 LAGS=1 MAVER=1
MODEL=IMPLICIT SIGMA=3
V12=3 V2=3 V3=3 V5
E1=V11-(V12*B1+V2*B2+V3*B3+V5*B4+B5)
(6F4.0)
 1.0 1.0 1.0 1.0 1.0 1.0
B6=1.
STOP
```

Note several things about this run. First, we "freeze" the value of the error variance at 1, and shut off its orthogonality condition (SIGMA = 3). Because this is a "regression", this will not affect the coefficient estimates, and can prevent estimation failure which sometimes occurs if sigma is driven to infinity. Second, it is necessary to explicitly list the variables which will be used as instruments i.e. orthogonal to the equation residuals, on the card after the equation. We use two lagged values of HOURS and WAGE but only contemporaneous values of EDUC since it is constant over time and EXPER since it is linear in time. Finally, note that in computing standard errors, we assume that residuals have the same order of correlation (MAVER) as we assumed for the lagged instruments (LAGS).

5.6. Large Data Sets and Models

Particular problems can arise with very large data sets or models which requires extra care in estimation, or large costs can arise. In this section we present an example of an estimation strategy which might be followed in estimating a large model on a large data set.

Suppose that instead of five years of data on 120 women we had ten years on 2000 women for a total of 20000 observations. Since we have ten variables, it would take approximately 200000 words of storage (10 x 2000) or 800k bytes on an IBM just to store the data. If we ran a model with all ten variables, it would take twice that amount, as run data would also have to be stored. Thus, the first suggestion would be that disk storage be used, and, if several different models are anticipated, a binary data file set up. The first step might be to look at variable means, do any desired transformations, and create this file.

```
OBSV=2000 TIME=10 INPUT=17 FORMAT=(10F6.0) VARI=10 PRINT=1 TRANS=2 $
      DISK=2
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
V11=[HOURS](+1)
V12=[AGE]-[EDUC]+6
HOURS+1 EXPER
STOP
```

JCL for device 18, the program's temporary disk area, should be set to save the data. All subsequent runs can use the file 18, and if PRINT is not specified, bypass the first part of the program.

The next step might be to take a subset of the observations to estimate the model on, making sure that the data is o.k., and providing starting values for the larger complete run.

```
OBSV=2000 TIME=10 INPUT=18 DISK=1 VARI=12
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE HOURS+1 EXPER
EQUA=2 TYPE=1 ITERA=20 UCROS=1TO100 UTIME=1TO3 CWRITE=1 START=2
DEP=V2 MODEL=TOBIT
V3 V6 V7 V8 V9 V10 V12
DEP=V4 MODEL=PROBIT
V3 V6 V7 V8 V9 V10 V12
STOP
```

Note that we "punch" or write out the coefficients (in format (5D16.9)) on device IPC (usually 7). Thus your JCL should specify device 7 as a keep data set. Note also that we are using only 300 of the 20000 observations, thus can keep the run data in core.

The next step is to run the entire model using the above starting values. Eventually we will want to run the equations together (TYPE=4) but for the moment we use the considerably cheaper TYPE=1 option. In the run that follows, we only allow for 10 iterations so that misspecified bad runs won't cost us a fortune, and we write out coefficients at each iteration (CWRITE = 2) in case the computer fails and we lose intermediate output. We copy the

starting values into our input file from the previously used device 7. Finally, because the run file is now large, we also use disk storage for the run data, requiring the temporary disk area device 19.

```
OBSV=2000 TIME=10 INPUT=18 DISK=1 VARI=12
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE HOURS+1 EXPER
EQUA=2 TYPE=1 ITERA=10 CWRITE=2 DISK=1
DEP=V2 MODEL=TOBIT
V3 V6 V7 V8 V9 V10 V12
DEP=V4 MODEL=PROBIT
V3 V6 V7 V8 V9 V10 V12
 0.123456789D+00 0.123456789D+00 0.987654321D+01 0.555666777D-01 ...
 0.617874362D-04 0.172378889D-01-0.678999977D+00 0.889977665D+02 ...
 0.454747474D+00 0.228828282D+00 0.778899900D+01 0.788899999D+03 ...
 0.467383992D+01 0.123466778D-01
STOP
```

(The ... are used because the manual width is not 80 columns).

If there is evidence from this run that things are working (gradients are falling and only 3 or 4 tries per iteration are required), then a final converging run could be put in. New starting values taken from the end of the first run should be substituted. Several other changes might also be useful. We usually bypass steepest descent since the program has already gone through it (ESTIMA = 2) and set only a gradient convergence criteria (CNVRG = 2). This can save considerable expense if the program essentially converges (gradients less than D-03 with scaled data) but has trouble in the final step (direction of search uphill etc.)

```
OBSV=2000 TIME=10 INPUT=18 DISK=1 VARI=12
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE HOURS+1 EXPER
EQUA=2 TYPE=1 ITERA=10 CWRITE=2 DISK=1 ESTIMA=2 CNVRG=2
DEP=V2 MODEL=TOBIT
V3 V6 V7 V8 V9 V10 V12
DEP=V4 MODEL=PROBIT
V3 V6 V7 V8 V9 V10 V12
 0.223451789D+00 0.223451789D+00 0.987154322D+01 0.555111777D-01 ...
 0.127874312D-04 0.272378889D-01-0.178999977D+00 0.889977115D+01 ...
 0.454747474D+00 0.228828282D+00 0.778899900D+01 0.788899999D+03 ...
 0.417383992D+01 0.223411778D-01
STOP
```

If these models converge, additional refinements might be explored using the final converged estimates as starting values. Standard errors might be estimated using the time-adjusted criteria.

```
OBSV=2000 TIME=10 INPUT=18 DISK=1 VARI=12
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE HOURS+1 EXPER
EQUA=2 TYPE=1 DISK=1 ESTIMA=4 PSTAN=1
DEP=V2 MODEL=TOBIT
V3 V6 V7 V8 V9 V10 V12
DEP=V4 MODEL=PROBIT
V3 V6 V7 V8 V9 V10 V12
 0.193351489D+00 0.193351489D+00 0.964153319D+01 0.534111444D-01 ...
 0.124843312D-03 0.242348889D-01-0.148999944D+00 0.889944115D+01 ...
 0.353434343D+00 0.228828282D+00 0.448899900D+01 0.488899999D+03 ...
 0.314383992D+01 0.223311448D-01
STOP
```

Alternatively the equations could now be jointly estimated (ML) using a starting value for the correlation provided by you (say .5).

```
OBSV=2000 TIME=10 INPUT=18 DISK=1 VARI=12
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE HOURS+1 EXPER
EQUA=2 TYPE=4 DISK=1 ESTIMA=2 ITERA=10 CNVRG=2 PSTAN=1
DEP=V2 MODEL=TOBIT
V3 V6 V7 V8 V9 V10 V12
DEP=V4 MODEL=PROBIT
V3 V6 V7 V8 V9 V10 V12
 0.193351489D+00 0.193351489D+00 0.964153319D+01 0.534111444D-01 ...
 0.124843312D-03 0.242348889D-01-0.148999944D+00 0.889944115D+01 ...
 0.353434343D+00 0.228828282D+00 0.448899900D+01 0.488899999D+03 ...
 0.314383992D+01 0.223311448D-01 0.500000000D+00
STOP
```

These runs only provide a few examples of the types of models estimable with **HotzTran** and a flavor of strategies for usage. These are likely to differ from problem to problem, and certainly from machine to machine.

6. SUBPROGRAM MinReg

Because users of **HotzTran** may occasionally want to use the front end of the program and data setup with simpler models, a simple regression subprogram is accessible within **HotzTran** which we refer to as **MinReg**. With a couple of exceptions, it uses data setup cards -- DATA CONTROL CARD, RUN TITLE CARD, VARIABLE NAMES CARD(S), and DATA TRANSFORMATION CARDS -- identical to those in the main **HotzTran** program.

MinReg is designed to run only regression models, without GLS transformations, using either raw data files or moment matrices. Only linear models can be run, and parameter restrictions cannot be imposed. That is, it can be used to estimate models of the form:

$$y_{it} = x_{it}'\beta + \delta_{it} \qquad (6.0.1)$$

The same data selection options available in the primary **Hotztran** program are available, and observations can be weighted and residual analysis performed. **MinReg** can be set up to loop over different samples, potentially rewinding files in between. Unlike the main **HotzTran** program, **MinReg** can be run entirely from moment matrix input. Moreover, the moment matrix can be saved, or read in using a user supplied FORTRAN subroutine, SUBROUTINE USEM.

MinReg is accessed within **HotzTran** by substituting the **MinReg** MASTER CONTROL CARD with the command TYPE = MINREG on it, in place of the normal MASTER CONTROL CARD. The data setup cards are the same as those used for normal **HotzTran** runs (although some additional data commands are available which only work with **MinReg**), and **MinReg** will work with the same data files as used in the rest of **HotzTran**. The general command rules for "cards" in **MinReg** are the same as those of the main program (80 column cards, $ separations, only the first two letters of a command are necessary).

Once in **MinReg**, there are three different types of **MinReg** cards which can be read as often as desired to run different models. These are: (1) the MASTER CONTROL CARD (**MinReg**), (2) the DEPENDENT VARIABLE CARD, and (3) the INDEPENDENT VARIABLE CARD.

The MASTER CONTROL CARD (**MinReg**) determines which observations are used for the regression(s), whether observations are weighted, whether coefficients are punched, if a default constant is set, if a Durbin-Watson statistic is computed, and if residual analysis is performed. Whatever options are selected they will hold until another MASTER CONTROL CARD (**MinReg**) is read. To use **MinReg**, the first card read by **MinReg** after the data setup cards or **ConTim** and/or **HotzTran** runs must be a MASTER CONTROL CARD (**MinReg**) with the command TYPE = MINREG on it. As long as the program remains in **MinReg**, subsequent MASTER CONTROL CARD (**MinReg**) cards do not have to have this command.

The DEPENDENT VARIABLE CARD contains a list of dependent variables which will be used for every regression which follows until another DEPENDENT VARIABLE CARD is read in. Variables can be listed individually or in strings.

The INDEPENDENT VARIABLE CARD is read for each separate regression and contains the list of independent variables for the regression. Variables can be listed individually or in strings. They can be referred to be name or number. The regression will be run for each of the dependent variables listed on the last DEPENDENT VARIABLE CARD. As many of these cards can be used as desired.

The program will stay in **MinReg** until it encounters the command STOP or reads a TRANSITION CARD or detects a MASTER CONTROL CARD of the main **HotzTran** program or **ConTim** by the command TYPE = 1 or 2 or 3 or 4 or **ConTim**. If it reads a TRANSITION card, it will exit **MinReg** to read another data file, and can only return when it encounters TYPE = MINREG on a MASTER CONTROL CARD. Similarly, if it reads a MASTER CONTROL CARD of **HotzTran** or **ConTim** it will exit to those subprograms.

The details of the control cards are given in the remainder of this chapter. Since the DATA CONTROL CARD, RUN TITLE CARD, VARIABLE NAMES CARD, and DATA TRANSFORMATION CARDS, are the same as the main **HotzTran** program, details of the commands are not repeated (see Chapter 3). We do briefly indicate those commands that differ slightly from the regular version.

6.1. DATA CONTROL CARD

The DATA CONTROL CARD commands determine the structuring of the input data, including data transformations, and to the choice of summary statistics printed out by the program. The commands VARI, OBSV, TIME, INPUT, PRINT, MISS, NAME, TRANS, and LINE are unchanged in the **MinReg** subprogram from the discussion given in Chapter 3. The only commands which are changed (expanded) are FORMAT and DISK. Their options are:

Option	Description

FORMAT = XXX

where XXX is explained below. The default is the integer 0. This option determines the input form of the data. Observations are read either one at a time, with VARI variables read per observation, as an entire observation matrix or as a pre-created moment (sums of squares and cross products) matrix.

- FORMAT = 0. With this option observations are read one at a time with a single precision binary read. If this option (cheapest) is used and no print is desired, the program will bypass the observation read section.

- FORMAT = MOMENT. This option implies that instead of raw data, a moment matrix is read by a double precision binary read. The matrix is read as a lower-triangular half matrix (i.e., (1,1); (1,2); (2,2); (3,1); etc.) with single read statement. The matrix should have a dimension of VARI plus ONE (thus (VARI+1)*(VARI+2)/2 elements are read), the last variable being constant. Thus, the last element of the matrix should be the number of observations. If this option is used, no transformations can be made (VAR* = VARI), all observations must be used for all runs and variables cannot be weighted. Observations cannot be printed, nor can residuals or Durbin-Watson statistics be computed.

- FORMAT = (...). If this option is selected observations will be read one at a time by the format listed after the equal sign. The format should be identical to a FORTRAN statement, and must begin and end with parenthesis, e.g., FORMAT = (7X,4F7.3). If the format continues for more than one line a $ break can be used.

- FORMAT = USER This option implies that observations will be read one at a time by the user-supplied SUBROUTINE USER. USER is a bare-bones subroutine which can be altered to do data transformations, etc.. It is called once for each observation. A listing of SUBROUTINE USER is given elsewhere in the manual.

- FORMAT = USEO This option implies that observations will be read as an entire observation matrix by the user-supplied SUBROUTINE USEO. USEO is a bare-bones subroutine which can be altered to do data transformations, etc.. It is called once. A listing of SUBROUTINE USEO is given elsewhere in the manual.

- FORMAT = USEM. This option implies that the moment matrix will be read by the user-supplied SUBROUTINE USEM. USEM is a bare-bones subroutine which can be altered as desired. It is called once. A listing of SUBROUTINE USEM is given at the end of this chapter. Similar restrictions apply to runs using FORMAT = USEM as FORMAT = MOMENT.

DISK = # where # is 0, 1, 2, 3, 4, or 5. The default is 0. This option determines internal disk usage.

- If DISK = 0 observations are stored in core (can take a lot of core). This option cannot be used with FORMAT=MOMENT or FORMAT=USEM.

- If DISK = 1 observations are assumed to be stored on a single-precision binary disk file already set up by the user. If this option is used, also use the option FORMAT = 0. The DISK = 1 option can only be used when no DATA TRANSFORMATION(S) are being done (i.e., TRANS must be equal to 0). If this option is used and PRINT = 0, the program will bypass the observations processing part of the program. This can save one read of the observation file.

- If DISK = 2 the program will write observations on the binary disk file INT (set to 18) and not use core storage. If INT is saved, it can be used for a subsequent run with the option DISK = 1. This option cannot be used with FORMAT=MOMENT or FORMAT=USEM.

- If DISK = 3, the program will store only the moment matrix of the entire sample. If this option is used all observations must be used for all regressions and variable weighting, residual analysis, and Durbin-Watson statistics cannot be computed.

- If DISK = 4, observations are stored in core (same as Option (=0)).

The moment matrix of the entire sample will be written on the binary disk file INT (set to 18) in double precision. If INT is saved, the moment matrix can be used for a subsequent run with the option FORMAT=MOMENT. This option cannot be used with FORMAT=MOMENT or FORMAT=USEM.

- If DISK = 5, the program will store only the moment matrix of the sample. Like option (=4), the moment matrix will also be written on a binary disk file INT. This option cannot be used with FORMAT=MOMENT or FORMAT=USEM.

6.2. MASTER CONTROL CARD (MinReg)

This card must be read first upon entry to **MinReg**. The first card must have the command TYPE = MINREG on it as well, although it need not be included on subsequent MASTER CONTROL CARD (**MinReg**)s. The card also determines which subset of the data sample is used for the regressions which follow. Once an option is specified on the MASTER CONTROL CARD (**MinReg**) (such as residual print), it will apply until another MASTER CONTROL CARD (**MinReg**) is read in. Each time a MASTER CONTROL CARD (**MinReg**) is read in all options revert to their defaults and must be respecified. As many MASTER CONTROL CARD (**MinReg**)s as desired can be read in. The program will stop when it encounters STOP or exit when it detects the MASTER CONTROL CARD for **HotzTran** or ConTim.

Option	Description
TYPE = MINREG	This command must be used on the first MASTER CONTROL CARD (**MinReg**) used in **MinReg**.
STOP	This command instructs the program to stop. It should be placed at the end of the control file. The entire command STOP must be spelled out (no abbreviations).
MISS = #	where # is 0, 1, or 2. The default is 0. This option determines the treatment of observations with missing data. It does not apply unless MISS is also specified on the DATA CONTROL CARD. If MISS = 0 (here) then no observations are excluded for missing data. If MISS = 1 then all variables needed for each separate regression (independent, dependent, and weight) are checked. If any have missing values the observation is excluded for that regression. This option requires that a separate moment matrix be computed for each regression (although only needed variables are summed). If MISS = 2 (here) then any observation with any missing variables is excluded from all regressions. This option does not require summing moments for each regression. Neither of the MISS options can be used if DISK = 3 or DISK = 5.
UTIME = $\#_1$ TO $\#_2$	where $\#_1$ is any integer from 1 to TIME and $\#_2$ is any integer from $\#_1$ to TIME (or the maximum number of time periods if TIME is variable). The default for $\#_1$ is 1 and for $\#_2$ is TIME. This option determines the range of time periods used for the run. This option cannot be used with DISK = 3 or

DISK = 5.

XTIME = $\#_1$ TO $\#_2$

where $\#_1$ is any integer from 1 to TIME and $\#_2$ is any integer from $\#_1$ to TIME (or the maximum number of time periods if TIME is variable). This option determines the range of time periods which will not be used in the run. It cannot be used with UTIME or if DISK = 3 or DISK = 5.

UCROS = $\#_1$ TO $\#_2$

where $\#_1$ is any integer from 1 TO OBSV and $\#_2$ is any integer from $\#_1$ to OBSV. The default for $\#_1$ is 1 and for $\#_2$ is OBSV. This option determines the range of cross-sectional observations used for the run. Any units not in the specified range will not be used . With non-panel data it determines which observations are used. This option cannot be used with DISK = 3 or DISK = 5.

XCROS = $\#_1$ TO $\#_2$

where $\#_1$ is any integer from 1 TO OBSV and $\#_2$ is any integer from $\#_1$ to OBSV. This option determines the range of cross-sectional observations which will not be used for the run. It cannot be used with UCROS or if DISK = 3 or DISK = 5.

UOBSV = V$\#_1$.XX.V$\#_2$ or UOBSV = V$\#_1$.XX.$\#_3$

where $\#_1$ and $\#_2$ are any integers from 1 to VAR*, $\#_3$ is any real number w/o scientific notation, and XX is one of the logical operators GT, LT, GE, LE, EQ, NE. There are no defaults. Variable names bracketed by [and] can be substituted for V$\#_1$ and/or V$\#_2$. If UOBSV is not specified all observations will be used. This option specifies a Fortran IF statement which determines which observations are used in a run. Any observations not meeting the criteria will not be used. V$\#_1$ specifies a variable number whose value is compared either to the value in variable V$\#_2$ or the real number, $\#_3$ using the comparison indicated by the logical operator (conventional fortran treatment, e.g., GT means greater than). If the comparison is true (i.e., V22.GE.3.27 or V27.EQ.V28 or [REEDER].NE.[NEW]) then the observation is used. The comparison is made separately for each time period of each cross-section unit. This option cannot be used with DISK = 3 or DISK = 5.

WEIGHT = V#

where # is any integer from 1 to VAR*. There is no default. This option determines if observations will be weighted in running the regressions which follow. The variable name bracketed by [and] can be substituted for V#. The variable whose number is specified will be used to weight the cross-products used in accumulating the moment matrix. This is equivalent to weighting observations by the square root of the variable. Any observations will non-positive weights will not be used. The constant term will also be weighted. Variable means listed on the print-out will be weighted means, and all run summary statistics will be adjusted. This option cannot be used if DISK = 3 or DISK = 5.

CWRITE = # where # is 0 or 1. The default is 0. This option determines if estimated parameters are punched (written on disk). If CWRITE = 0 nothing is punched.

If CWRITE = 1 parameters are punched in (5D16.9) format on device IPC. This device is set within **MinReg**, generally to 7.

RPRINT = # where # is 0 or 1. The default is 0. This option determines whether the program performs and prints a residual/predicted value analysis. If RPRINT = 1, the program will compute predicted values and residuals for each observation/time period analyzed. If RPRINT = 0 no residual analysis is done. This option cannot be used if DISK = 3 or DISK = 5.

RWRITE = # where # is any non-negative integer. The default is 0. This option determines if residual analysis information is punched/written on disk for each observation/time period/regression. This option can be used independently of RPRINT. If RWRITE = 0 nothing is punched. If # > 0 then # will be the device number that residual terms are punched/written on. Order of punch is the same as the residual print with a format of (3D16.9) (dependent variable, predicted value, and residual). This option cannot be used if DISK = 3 or DISK = 5.

MWRITE = # where # is any non-negative integer. The default is 0. This option determines if the covariance matrix of estimated coefficients is punched/written on disk. If MWRITE = 0 nothing is punched. If # > 0 then # will be the device number that the covariance matrix terms will be punched/written on. The matrix is punched as a single vector in (5D16.9) format. Element order is (1,1), (2,1), (2,2), (3,1), (3,2), (3,3), (4,1), ...

AUTO = # where # is 0 or 1. The default is 0. This option determines if a Durbin-Watson statistic and predicted autoregressive ρ are computed for each regression. It is valid only if TIME > 1 and properly adjusts for panel data. If AUTO = 1 the statistic is computed, otherwise it is not. This option cannot be used if DISK = 3 or DISK = 5.

CONST = # where # is 0 or 1. The default is 0. This option determines if a constant is used in each regression. If CONST = 0, it is. If CONST =1 it is not. This option can be overridden on any regression by adding a constant as variable V0 or CONSTANT or taking one out with NV0 or [CONSTANT].

6.3. DEPENDENT VARIABLE CARD

This card must be included at least once preceding any INDEPENDENT VARIABLE CARDS. It contains a list of the dependent variables to be used for each of the regressions which follow. The dependent variables will be used for all regressions which follow until another DEPENDENT VARIABLE CARD is encountered. The first dependent variable should be listed as: DEP=V# where # is the number of the variable. Variable names can also be used (brackets are not necessary). All subsequent dependent variables on the "card" should not have DEP= , only the V#. Variable strings can also be used, indicated by a "-". Thus, DEP = $V\#_1$ - $V\#_2$ will imply that all variables from $\#_1$ to $\#_2$ will be used. If variables (including those in strings) are preceded by an "N" they will not be be used. Thus, for example DEP=V1-V10.NV8 states that all variables between one and ten except eight are used. When used after "N", variable names must be bracketed with [and]. Variables (or strings) should be separated by blanks or commas and $ used for card continuation.

6.4. INDEPENDENT VARIABLE CARD

These cards contain the regression independent variables. As many regressions as desired can be run. Each regression will be run for each of the dependent variables listed on the last preceding DEPENDENT VARIABLE CARD. All variables must be listed, except the constant if CONST = 0. Variables are listed as V#, where # is the number of the variable. Variable names can also be used (brackets are not necessary). Variable strings can also be used, indicated by a "-" between variables. Thus $V\#_1$ - $V\#_2$ implies that all variables from $\#_1$ to $\#_2$ will be included in the regression.

If variables (including those in strings) are preceded by an "N", they will not be included. Thus, for example, V1-V10,NV8 implies that all variables from one to ten except variable eight will be included. If a variable appears more than once its status will be determined by the last reference. When used after "N", variable names must be bracketed with [and]. The constant command, CONST, can be overridden by adding or subtracting a constant as variable V0. It cannot be included in a string, however. Variables (or strings) must be separated by blanks or a comma.

Variables can be placed in any order on the EQUATION VARIABLE CARD. Coefficient order for print and punch will be determined by the order variables are listed on the INDEPENDENT VARIABLE CARD, followed by the constant (if used). **Note, that this differs from the ConTim or HotzTran rules**, where variable coefficients are ordered in ascending order according to their numbers, regardless of their order on the input card. The **MinReg** option is offered as more flexible, because **MinReg** does not have to contend with multiple equation systems.

6.5. TRANSITION CARD

This card signals the end of **MinReg** and the desire to read another data file. It has the following commands

Option	Description
SAMPLE	This command returns to the beginning of the program to read another DATA CONTROL CARD. If SUBROUTINES USER, USEO, or USEM are used, the variable ISAMP will indicate the number of the sample being read. It can be accessed by including the statement: COMMON /G/ISAMP in the subroutine. The entire command SAMPLE must be spelled out (no abbreviations).
REWIND = #	where # is any positive integer. This option will rewind the file on device #. The entire command REWIND must be spelled out (no abbreviations).
ENDFILE = #	where # is any positive integer. This option will end file the file on device #. The entire command ENDFILE must be spelled out (no abbreviations).

6.6. SUBROUTINES USER, USEO and USEM

There are three user-supplied subroutines which can be expanded by users of the **MinReg**

subprogram to read in observations one-at-a-time (SUBROUTINE USER) or as an entire observation matrix (SUBROUTINE USEO) or as a pre-created moment matrix (SUBROUTINE USEM). Code for USER and USEO are given in Chapter 3. SUBROUTINE USEM has the following skeletal form:

```
C     SUBROUTINE USEM
C     XX IS THE LOWER TRIANGULAR MOMENT MATRIX TO BE FILLED.
C     NVAR IS THE TOTAL # VARIABLES (EXCLUDING THE CONSTANT
C     WHICH IS LAST).  INN IS THE INPUT DEVICE.  IDD SHOULD
C     BE SET TO 1 IF AN INPUT ERROR OCCURS.  THE DIMENSION
C     OF XX IS (NVAR+1)*(NVAR+2)/2.
      SUBROUTINE USEM(XX,NVAR,INN,IDD)
      DOUBLE PRECISION XX(1)
      RETURN
      END
```

6.7. MinReg Examples

To illustrate the **MinReg** section of the program, consider the following set of simple examples. Suppose that the same data set is used that described in Chapter 5, consisting of 120 women observed for 5 time periods, with data on labor force participation, hours worked, wages, fertility, education, family income, age, kids, and race. To run a simple wage equation on several subsets of variables use:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=MINREG
DEP=[WAGE]
V4-V10
V4-V9
V5 V6 V8 V9
EDUC INCOME KIDS
STOP
```

To read a new sample using 1200 new observations off of device 18 and run the same equations with Durbin-Watsons, and residual analysis for the last equation only, run:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=MINREG
DEP=V3
V4-V10
DEP=V2-V3
V4-V9
V5 V6 V8 V9
EDUC INCOME KIDS
SAMPLE
OBSV=240 TIME=5 INPUT=18 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=MINREG AUTO=1
DEP=V3
V4-V10
DEP=V2-V3
V4-V9
V5 V6 V8 V9
AUTO=1 RPRINT=1
DEP=V3
EDUC INCOME KIDS
STOP
```

To run models for both hours and wages, to write the parameter covariance matrix on device 8. to use only time periods 2 to 4, to "punch" coefficients on device 7 for the first two equations only, and then run a probit in the main **HotzTran** program using the same data, run:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(10F6.0) VARI=10
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=MINREG MWRITE=8 UTIME=2TO4 CWRITE=1
DEP=V3 V2
V4-V10
V4-V9
UTIME=2TO4 MWRITE=8
DEP=V2-V3
V5 V6 V8 V9
EDUC INCOME KIDS
TYPE=1 START=2 ITERA=20
DEP=[LFP].GT.0 MODEL=PROBIT
FERT EDUC AGE KIDS RACE
STOP
```

7. SUBPROGRAM ConTim

The primary focus of the **HotzTran** program is on the use of qualitative dependent variable models in a variety of settings. The particular model forms we allow -- Probit, Tobit, Double Tobit, logit, truncated, regressions, and implicit functions -- are all single equation models, and with the exception of Logit, model forms with latent normal residual forms. The simple equation form of these models fit nicely into the non-linear, multiple equation, multiple time period forms allowed in **HotzTran**. However, these model forms by no means exhaust the list of commonly used qualitative dependent variables. Missing in particular are the logistic multiple choice models and recent choice model forms proposed for continuous time settings. The **HotzTran** subprogram **ConTim** is designed to allow **HotzTran** users to run other qualitative dependent variable models using the data setup form and command language of **HotzTran**.

The **ConTim** subprogram allows the estimation of two distinctly different types of models. The first is the general class of logistic choice models. These include multinomial logit, conditional logit, ordered logistic, ordered probit (N-Chotomous), tri-chotomous (3 choice) probit, and poisson regression models. The second model type estimable in **ConTim** includes the general class of continuous time markov-type or proportional hazard models. These can be estimated in both continuous time or discrete form. Since both of these general models types concern estimation of processes in which the dependent variable is a "choice", it turns out to be relatively convenient to allow estimation of both types within the same program.

Although the program controls and general estimation procedures used in **ConTim** are virtually the same as the main **HotzTran** program, there are differences. As indicated above, the model forms are also different. For these reasons, we divide the remainder of the Chapter into several sections. In the next section, we provide a brief write-up of the logistic and probit choice models available in **ConTim**. In the following section we provide an equally terse discussion of the continuous time models available. Finally in the remainder of the chapter, we discuss the **ConTim** control cards and commands.

7.1. Choice Models

The **ConTim** section of **HotzTran** can be used to estimate a variety of choice models. These are models where the "dependent" variable only takes on discrete, ordinal, values. This class of model is suitable for processes like transportation mode, or occupation, or political affiliation which cannot easily be described by a cardinal variable. Within the general class of choice models, there are a multitude of different specifications, a variety of which are estimable by **ConTim**. The topic is too large to do justice to here; but, in the remainder of this section, we try to present a brief overview of the models which are available. For a more complete discussion, the user is referred to a general reference such as Maddala (1983), McFadden (1974), or Altman et al. (1981).

There are four different choice models which can be estimated in **ConTim** -- (1) unordered logistic choice models, (2) unordered 3 choice probit models, (3) ordered logistic or probit models, and (4) poisson regression models. We begin our discussion with the unordered logistic choice models.

7.1.1. Unordered Logistic Choice Models

Consider a process, where the dependent variable for the n^{th} observation, y_n, takes on one of k discrete values (such as choice of occupation). For all unordered logistic choice models, we assume that the probability of choice i being chosen given a vector of independent variables, x_n, is:

$$P(\text{Choice } i \mid x_n) = \frac{\exp(x_{ni}' \beta_i)}{\Sigma_{j=1,k}\exp(x_{nj}' \beta_j)} \tag{7.1.1}$$

This is the standard logistic model, where the probability of a choice is equal to the exponentiated "expected value" of the choice divided by the sum of the expected values for all choices. The expected value of each choice is a function of exogenous variables and β weightings. Note that both the x's and the β's are subscripted by choice. This implies that both the relevant x variables and their weightings may differ by choice. We assume that all k choices are available for each observation although this assumption can be relaxed.

The general logistic choice model in equation (7.1.1) is not estimable without normalizations and/or parameter restrictions. The most commonly seen restriction is to assume that the x's are subscripted by choice, but the β's are not. Commonly referred to as the *Conditional Logit Model* (see McFadden(1974)), the choice probabilities become:

$$P(\text{Choice } i \mid x_n) = \frac{\exp(x_{ni}' \beta)}{\Sigma_{j=1,k}\exp(x_{nj}' \beta)} \tag{7.1.2}$$

This type of model arises quite naturally in cases where the x's represent characteristics of the choices, not the decision maker. The β vector, which is constant across choices, represents weighting of variables assigned by the decision maker. The conditional logit model can be run in **ConTim** with the command, MODEL = CONDLO. Several options are available.

Since the choice probability in equation (7.1.2) is a ratio, the numerator and denominator can be divided by any constant without changing the probabilities. Thus, if we divide top and bottom by any choice value, $\exp(x_{ni}'\beta)$, the probabilities are unaffected. Note, that for choice i, the adjusted value would be $\exp(0) = 1$, and for other choices, $\exp([x_{nj}' - x_{ni}']\beta)$. Thus only relative x's count, and any x variable which is constant across choices (such as any characteristic of the decision maker or a constant) would drop out. Since there is only one β vector, the numbering of the choices is arbitrary, and each observation could be normalized by a different choice. In **ConTim**, the program can do the normalization, or the user can read in normalized data (see TDATA). In this case, it is assumed that the ratio for each observation is normalized by the x's of the observation's "chosen" choice, i_n, and only relative x's, $x_{nj}-x_{ni}$, of the other k - 1 "inferior" (unchosen) choices are read in. It is also possible to use a variable choice set size (DSIZE).

The *Multinomial Logit Model* (see Nerlove and Press (1973)) is normalized in the opposite way to the conditional logit model. Here, it is assumed that the x's are constant across choices, but that each choice has a different weighting vector:

$$P(\text{Choice } i \mid x_n) = \frac{\exp(x_n' \beta_i)}{\Sigma_{j=1,k} \exp(x_n' \beta_j)} \qquad (7.1.3)$$

This model, invoked by MODEL = MULTLO, arises most naturally in cases where the x's represent characteristics of the decision maker, not the choices, and the β's represent different weightings of these x's depending on the choice. Like the conditional logit model, the multinomial logit model is identified only up to scale. Any arbitrary vector, γ, can be added to each β without affecting probabilities. **ConTim** allows two different normalizations (see CSCALE) -- setting the β vector for choice one to zero (thus all coefficients are relative to choice one), or setting the coefficients to sum to zero for each x variable (choice one's coefficient is set equal to the negative of the sum of the corresponding coefficients for the other choices). For the multinomial logit model, the choice numberings are not arbitrary as they are for the conditional logit model. For this reason it is inadvisable to use variable choice set sizes although it is allowed (see DSIZE).

Some models are a mixture of the conditional and multinomial logit models, with some x's and some β's varying by choice. **ConTim** allows the estimation of these models with the command MODEL = FULLLO. Expected values in this case are set separately by the user for each choice, the probabilities of the various choices can depend upon different sets of x's, and the probabilities can be functions of $x'\beta$, i.e., linear in the x's and β's or of a wide class of nonlinear functions, i.e., $g(x_n, \beta_i)$, which are specified by the user USING the NON-LINEAR EQUATION CARDS. Special care should be made to make sure that the model is fully identified and normalized. Failure to do this will be signaled by convergence failure or coefficients driven to infinity.

With two choices, both the multinomial and conditional logit models reduce to the binary logit model available in the main program **HotzTran**. In the estimation of all the unordered logistic models, the log-likelihood is just the log of the probability of the chosen choice.

7.1.2. Unordered 3 Choice Probit Model

ConTim allows the estimation of a three choice *Multinomial Probit Model* with the command MODEL = TPROB (see Hausman and Wise (1978)). Underlying the Logistic choice models is "the axiom of independence of irrelevant alternatives," which implies that unobserved components of choice "values" will be uncorrelated. This assumption is unnecessary for the multinomial probit model. Because of computational problems, however, the model can only be estimated for precisely three choices. In **ConTim**, the user specifies three "value" functions of the form, $x_{in}'\beta_i + \epsilon_{in}$. The covariance of the i^{th} and j^{th} ϵ's is assumed to be σ_{ij}. With this specification, the choice probabilities are given by:

$$\begin{aligned} P(\text{Choice } 1 \mid x_n) &= F(D_{12}, D_{13}, \rho_1) \\ P(\text{Choice } 2 \mid x_n) &= F(-D_{12}, D_{23}, \rho_2) \\ P(\text{Choice } 3 \mid x_n) &= F(-D_{13}, -D_{23}, \rho_3) \end{aligned} \qquad (7.1.4)$$

where F( ) is the standard bivariate normal distribution function with correlations,

$$\rho_1 = (\sigma_{11} - \sigma_{13} - \sigma_{12} + \sigma_{23})/[(\sigma_{11} + \sigma_{22} - 2\sigma_{12})(\sigma_{11} + \sigma_{33} - 2\sigma_{13})]^{1/2}$$

$$\rho_2 = (\sigma_{22} - \sigma_{12} - \sigma_{23} + \sigma_{13})/[(\sigma_{11} + \sigma_{22} - 2\sigma_{12})(\sigma_{22} + \sigma_{33} - 2\sigma_{23})]^{1/2}$$

$$\rho_3 = (\sigma_{33} - \sigma_{13} - \sigma_{23} + \sigma_{12})/[(\sigma_{11} + \sigma_{33} - 2\sigma_{13})(\sigma_{22} + \sigma_{33} - 2\sigma_{23})]^{1/2}$$

and bounds,

$$D_{12} = (x_{1n}'\beta_1 - x_{2n}'\beta_2)/[\sigma_{11} + \sigma_{22} - 2\sigma_{12}]^{1/2}$$

$$D_{13} = (x_{1n}'\beta_1 - x_{3n}'\beta_3)/[\sigma_{11} + \sigma_{33} - 2\sigma_{13}]^{1/2}$$

$$D_{23} = (x_{2n}'\beta_2 - x_{3n}'\beta_3)/[\sigma_{22} + \sigma_{33} - 2\sigma_{23}]^{1/2}$$

As written, the model requires several normalizations before it could be estimated. With common x's and no parameter restrictions, at most three of the covariance terms -- σ_{11} .. σ_{23} -- are identified. The user can control the normalization. However, the program's default is to set all three variance terms to one. If the x's vary by choice, but the β's do not (like the conditional logit model), this makes sense as the choices may have no inherent meaning. The option CSTAN controls the normalization.

The program requires that the multinomial probit model be fully specified by the reader (as with FULLLO). Non-linear specifications are allowed as well. The log-likelihood is given by the log of the probability of the chosen choice.

7.1.3. Ordered Logistic and Probit Models

The choices of the unordered logistic and probit models discussed above will generally have no inherent ordering. In some problems, however, there is a natural ordinal (though not cardinal) ranking of choices. This might arise, for example, in a model of bond ratings where the the user did not want to impose a cardinal scale on the ratings, but wants to take account of the knowledge that a AAA rating is "better" than AA. Both the *Ordered logistic* and *Ordered Probit* (sometimes called N-chotomous) models have this feature. In both models, it is assumed that k discrete ranked choices are available to the decision maker, and that choices are determined by the latent unobserved variable y^*, where:

$$y_n^* = x_n'\beta + \epsilon_n \tag{7.1.5}$$

The observed choice is determined by estimated "thresholds", μ, with:

Choice 1 if $y_n^* < \mu_1$

Choice 2 if $\mu_1 < y_n^* < \mu_2$

.
.

Choice k if $\mu_{k-1} < y_n^*$

where 1 is the lowest choice, and k is the highest. It is assumed that the choice selected is observed, but that y^* is not.

Since the latent variable is unobserved, its mean and scale have to be normalized. In **ConTim**, μ_1 is set to zero, and for the ordered logistic model (MODEL = ORDLO), ϵ is assumed to be standard logistic, and for the ordered probit model (MODEL = NPROB), ϵ is assumed to be standard normal.

For the ordered logistic model, the model can be represented by a sequence of binary logistic models determining the probability that an observation will have a choice ranking below (one is lowest) or equal to each choice. Here the probability of drawing a choice below or equal to choice i is the binary logistic:

$$P_{in} = \frac{1}{1+\exp(-x_n'\beta - \mu_i)} \tag{7.1.6}$$

The probability of choice i given x_n is then $P_{in} - P_{i-1,n}$ (for choice one it is P_{1n}).

The ordered probit model probabilities can be derived and used similarly, except that:

$$P_{in} = F(x_n'\beta + \mu_i) \tag{7.1.7}$$

where F( ) is the standard cumulative normal distribution function. For both models, the probability of choice k is given by one minus the sum of the other probabilities.

The log-likelihood of the ordered choice models is given by the log of the chosen choice probabilities. In estimation users should be sensitive to violations of the orderings. The program will estimate the thresholds, μ_2 ... μ_{k-1}. If in fact the choices are not truly ordered in this manner, the program may fail to converge, or μ estimates get very close to one another, or the regression constant (which should be used) become very large. The ordered choice models have few **ConTim** options, except the possibility of non-linear equations.

7.1.4. Poisson Regression

The Poisson regression model (MODEL = POISSON) represents another form of ranked or ordered choice model. Generally in this case, however, the process of interest is not one which is categorical, but which is discrete none-the-less (see Maddala (1983)). Examples might be the number of flaws in a piece of wire, or sales in a given time period, or absentee days in a year. Clearly, these processes could be modeled as regressions with a standard dependent variable. The poisson regression model offers a slight variant which may be attractive when the number of likely occurrences during the unit of observation is small.

For the poisson regression model, we assume that the number of occurrences of an event, y_n, for the n^{th} observation during a fixed time period is generated by a poisson process where:

$$\text{Prob}(y_n = r) = \frac{\exp(-\lambda_n)(\lambda_n)^r}{r!}$$

where r takes on the values -- 0, 1, 2, ... Furthermore, we assume that the mean of the poisson process is given by:

$$\ell n(\lambda_n) = \ell n(c_n) + x_n'\beta$$

where c_n is the length of the time period. If all observations have the same length time period (or number of repetitions) then the term for c_n will be incorporated in the constant (see DSUMS).

The poisson regression model is almost identical to the semi-log-linear regression model. The major difference is that it imposes the restriction that the regression mean and error variance be the same which is embodied in the poisson specification. In **ConTim**, the poisson model is estimated by iterative ML.

7.2. Continuous Time Models

ConTim can be used to estimate a general class of continuous time, multi-state *Markov* and *Proportional Hazard* models by the command MODEL = HAZARD. Unlike all other models estimable in **HotzTran** these models are defined in continuous time. At any given instant, an observation is defined as being in a state. The model defines the instantaneous probabilities of the observation "transiting" from the state it is in to one or more other states. Though a model can have many states, it need not be the case that an observation can transit from one state to all others.

This model form is quite convenient in describing processes where changes are very rare, and where the probability of "no change" at any given instant is virtually one. For example, state one might be "not having a child", and state two "giving birth." Obviously giving birth is rare. A common variation is the two-state *Failure Time* model, where state one is no failure, and state two would be failure (such as breakdown of a machine). In this case it

would be impossible to transit from state two to state one. An example where transit from each state to all others is possible would be a model of occupational choice.

Proportional Hazard models are thoroughly discussed in references such as Kalbfleisch and Prentice (1980), Flinn and Heckman (1982, 1983), and Avery and Hotz (1984) to which the reader is referred. Below we give only a brief description of the models as they are used in **ConTim.**

At any given point in time, τ, an observation, n, is defined as being in a state, i. Suppose that the current spell in state i began in period τ_i. We define the instantaneous probability that the observation will transit or change to another state, j, after t periods as the Hazard or escape rate, $h_{nij}(t)$. For Markovian models in **ConTim**, the hazard function for a transition to state j after t periods in state i (for all possible transitions, ij) is an exponential function of a set of exogenous and time-invariant x's:

$$h_{nij}(t) = \exp(x_{nij}' \beta_{ij}). \tag{7.2.1}$$

This is the hazard function for the ij^{th} transition. Note that we potentially allow both the x's and the β's to vary by transition. In addition, the x's can change over the duration of the spell in state i. In this case the hazard rate at duration t would be of the form:

$$h_{nij}(t) = \exp(x_{nij}(\tau_i+t)' \beta_{ij}). \tag{7.2.2}$$

Obviously (7.2.1) is just a special case of (7.2.2).

If there are a total of k states, the probability of remaining in state i, for more than T periods (i.e., of surviving in state i) is given by:

$$S_{ij}(T) = \exp\left\{ -\int_0^T \Sigma_{j=1\ j\neq i}^{k} h_{nij}(x_{nij}(\tau_i+t)' \beta_{ij})\ dt \right\}$$

which is just the survivor function. We assume that two different transitions cannot occur at the same time.

ConTim offers several more complicated options, however, with what is termed duration dependence. That is, we want the transition probabilities to depend upon how long an observation has been in a state. A Weibull model is one case which allows for such time dependence and its hazard function has the form:

$$h_{nij}(t) = \exp(x_{nij}(\tau_i+t)' \beta_{ij}) t^{\gamma_{ij}} \tag{7.2.3}$$

where t is the length of time the observation has been in the state. Weibull time dependence

can also be represented by adding a variable $\ell n(t)$ to the x's in the hazard. Another similar form of time dependence is the Gompertz form which is given by:

$$h_{nij}(t) = \exp(x_{nij}(\tau_i+t)'\beta_{ij})\exp(t\gamma_{ij}) \tag{7.2.4}$$

This is equivalent to adding the trend variable, t, to the x vector. **ConTim** also allows the more general and flexible Box-Cox specification of duration dependence with the hazard function suggested by Flinn and Heckman (1982).

$$h_{nij}(t) = \exp[x_{nij}(\tau_i+t)'\beta_{ij}]\ \exp[\left(\frac{t^{\lambda_{1ij}}-1}{\lambda_{1ij}}\right)\gamma_{1ij} + \left(\frac{t^{\lambda_{2ij}}-1}{\lambda_{2ij}}\right)\gamma_{2ij}], \tag{7.2.5}$$

ConTim allows all three types of time dependence to be estimated with the command DURA. The λ parameters in the Box-Cox function can be estimated or fixed by the user (allowing general polynomials in time). Moreover, the program takes advantage of the separability between the x portion of the hazard (which is constant within what we define as a spell) and the time dependent portion (which is not).

Since data do not come in continuous time form, to operationalize the these hazard models, it is necessary to make several approximations. We assume that all data come in the form of spells, where a spell is defined as a time period of length T, during which time the x's do not change, and at most one transition takes place. If the observation is in state i, and no transition takes place (i.e., an incomplete spell), the log-likelihood of the n^{th} spell is:

$$\ell n(L) = -\int_0^T [\Sigma_{j=1\ j\neq i}^{k} h_{nij}(t)]dt \tag{7.2.6}$$

If a transit to state m takes place (i.e., a complete spell), we generally assume it takes place at the end of the spell at duration T. The log likelihood is then:

$$\ell n(L) = x_{nim}(\tau_i+T)'\beta_{im} - \int_0^T [\Sigma_{j=1\ j\neq i}^{k} h_{nij}(t)]dt \tag{7.2.7}$$

These likelihoods are exact. In **ConTim** we use this form, but approximate the time integral by a Riemann Sum. As noted, we assume generally that the transition takes place at the end of a spell. If the transition timing is unknown, and spells are of different length (so it matters), then with most models (see KNOWN) the log-likelihood of a spell with a transition from state i to m is (substituting the Riemann sum):

$$\ell n(L) = x_{nim}(\tau_i+T)'\beta_{im} - \ln[\Sigma_{j=1\ j\neq i}^{k} h_{nij}(T)] + \ln[1 - \exp(-\Sigma_{t=0}^{T}\ \Sigma_{j=1\ j\neq i}^{k} h_{nij}(t))] \tag{7.2.8}$$

In these examples, the x's are constant within a spell. Thus, both the integral and/or the Riemann sum can be simplified to T times the hazard. The log-likelihood of a transition

from state i to m with Gompertz time dependence, for example, simplifies to:

$$\ell n(L) = x_{nim}(\tau_i+T)'\beta_{im} - \Sigma_{j=1\ j\neq i}^{k}\ [\exp(x_{nij}(\tau_i+t)'\beta_{ij})\Sigma_{t=0}^{T}\ \exp(t\gamma_{ij})] \qquad (7.2.9)$$

where n indexes the spell. This can be easily evaluated with only the last portion depending upon t.

ConTim also allows the time dependence terms to depend upon interactions with x variables. Thus, for example, the last term in equation (7.2.9) might be, $\exp(t\gamma_{1j} + z_n t\alpha_{ij})$, where z_n is one of the exogenous variables and α_{ij} is an estimated parameter. Another **ConTim** time dependence option is the ability to have the start in a state not coincide with the beginning of the spell (see DBEFOR).

Another option available in **ConTim** is to allow unobserved forms of heterogeneity to enter the proportional hazard model (see HETERO). The hazard is allowed to have the form:

$$h_{nij}(t) = \exp(x_{nij}(\tau_i+t)'\beta_{ij} + \alpha_{ij}v_n) \qquad (7.2.10)$$

where n indexes an "individual" (or cross-sectional unit) not a spell. The v term is a random effect, and the parameters α are estimated factor loadings. The key to estimation is the assumptions made about the distribution of v. Currently **ConTim** only allows v to be distributed with a gamma distribution with density function:

$$f(v) = [\Gamma(\theta_1)\theta_2^{\theta_1}]^{-1}v^{(\theta_1-1)}\exp[-v/\theta_2], \qquad (7.2.11)$$

where $v > 0$ and where θ_1 is the shape parameter and θ_2 is the scale parameter. In **ConTim**, θ_2 is normalized to 1 since scale can be captured in the estimated factor loadings, α_{ij}. The θ_1 shape parameter must be read in by the user (see GAMMA) and cannot be estimated.

When using data in panel form, **ConTim** will treat each time period observation as a "spell", and the cross-section units as "individuals". Thus multiple spells per individual could be used. The variable time option (see TIME) may be useful, therefore, if the number of spells varies across individuals.

Technically with k states, there are k(k-1) possible transitions. Each of these transitions could be governed by a separately estimated hazard function. In some models, however, some of these transitions may by "forbidden" or cannot take place. Similarly, users may desire to have the same hazard govern several different transitions. For this reason, **ConTim** allows the specification of a user-defined (see TRANS) number of hazards or transition probability functions. On a separate card (see the TRANSITION VECTOR CARD), the user defines which transitions each hazard applies to. The hazard functions or transition probabilities of the various transitions can depend upon different sets of x's, and they can be functions of $x'\beta$, i.e., linear in the x's and β's or of a wide class of nonlinear functions, i.e., $g(x_n, \beta_i)$, which are specified by the user USING the NON-LINEAR EQUATION CARDS. It is also possible

to have the x variables be used for all transition vectors (like the multinomial logit model), or be read in separately for each estimated transition function (see NDATA).

The proportional hazard model can also be used for discrete time models. Unlike the continuous time model, a spell represents a single, discrete, transition period. The probability of a transition from state i to state j in the spell n is given by:

$$P_{nij} = \exp(x'_{nij}\beta_{ij}) \tag{7.2.12}$$

for the exponential probability model (MODEL = EXP) and,

$$P_{nij} = \exp[\exp(x'_{nij}\beta_{ij})] \tag{7.2.13}$$

for the double exponential model (MODEL = DBLEXP). In each case the transition from i to i is computed as 1 minus the sum of probabilities for i to all other states. The double exponential model is most likely to yield parameter estimates close to those of a comparable continuous time model.

The discrete versions of EXP and DBLEXP can be used similarly to HAZARD except that time dependence and heterogeneity are not allowed. They can also be used in a manner similar to the choice models MULTLO or CONDLO. If all spells are defined as starting in state one (arbitrarily), and the k states represent choices, the estimated transition vectors (1,2), (1,3), (1,4) ... would have a similar interpretation to the coefficient vectors estimated for MULTLO. If only one vector were used to estimate all transitions the model would be similar to CONDLO.

As with the continuous time models, the discrete time hazard functions or transition probabilities of the various transitions can depend upon different sets of x's, and they can be functions of $x'\beta$, i.e., linear in the x's and β's or of a wide class of nonlinear functions, i.e., $g(x_n, \beta_i)$, which are specified by the user USING the NON-LINEAR EQUATION CARDS.

7.3. ConTim Program Controls

ConTim is accessed within **HotzTran** by substituting the **ConTim** MASTER CONTROL CARD with the command TYPE = CONTIM on it, in place of the normal MASTER CONTROL CARD. The data setup cards are the same as those used for normal **HotzTran** runs, and **ConTim** will work with the same data files as used in the rest of **HotzTran.** The general command rules for "cards' in **ConTim** are the same as those of the main program (80 column cards, $ separations, only the first two letters of a command are necessary).

Once in **ConTim**, there are seven different types of **ConTim** cards which can be read as often as desired to run different models. These are: (1) the MASTER CONTROL CARD (**ConTim**), (2) EQUATION VARIABLE CARDS(S) (optional), (3) NON-LINEAR EQUATION CARD(S) (optional), (4) the TRANSITION VECTOR CARD (optional), (5) RESTRICTION CARD (optional), (6) FORMAT CARD (optional), AND (7) COEFFICIENT STARTING VALUE

CARD. These cards are repeated (in order) for each "run" or "model". When no more models are desired, the program can be terminated by the command STOP, or alternatively return to the main **HotzTran** program to read a new data file (TRANSITION CARD) or estimate models in **HotzTran** or **MinReg**.

Many of the options are best explained in the write-up of the individual options. Almost all program specifications are determined on the MASTER CONTROL CARD (**ConTim**). This card determines which model-type is run, which observations are selected, how run convergence is determined, whether models are linear or non-linear, whether parameters are restricted, how starting values are obtained, whether observations are weighted or residuals printed, how time dependence is treated in continuous time models, coefficient scaling, and which variables determine equation "choices" (the dependent variables). To use **ConTim**, the first card read by **ConTim** after the data setup cards (or **MinReg** and/or **HotzTran** runs) must be a MASTER CONTROL CARD (**ConTim**) with the command TYPE = CONTIM on it. All subsequent MASTER CONTROL CARD (**ConTim**)s must also have this command (unlike **MinReg**). If the program detects a **MinReg** or **HotzTran** MASTER CONTROL CARd it will exit to those subprograms.

The specification of the model(s) independent variables is done on the EQUATION VARIABLE CARD(S) if the model is linear, or the NON-LINEAR EQUATION CARD(S) if the model is non-linear. For continuous time models, variables can be selected to automatically interact with time duration parameters, as well as enter the model linearly.

The TRANSITION VECTOR CARD is used with multi-state continuous time Markov models to indicate which transitions the estimated transition probability vectors apply to. Finally, the RESTRICTION CARD, FORMAT CARD and COEFFICIENT STARTING VALUE CARDs read in parameter restrictions and the read format and values respectively of the starting values of parameters.

The program will stay in **ConTim** until it encounters the command STOP or reads a TRANSITION CARD or detects a MASTER CONTROL CARD of the main **HotzTran** program or **MinReg** by the command TYPE = 1 or 2 or 3 or 4 or **MinReg**. If it reads a TRANSITION card, it will exit **ConTim** to read a new data file, and can only return when it encounters TYPE = CONTIM on a MASTER CONTROL CARD. Similarly, if it reads a MASTER CONTROL CARD of **HotzTran** or **MinReg** it will exit to those subprograms.

The details of the control cards are given in the remainder of this chapter. Since the DATA CONTROL CARD, RUN TITLE CARD, VARIABLE NAMES CARD, and DATA TRANSFORMATION CARDS, are the same as the main **HotzTran** program, details of the commands are not repeated (see Chapter 3).

7.4. MASTER CONTROL CARD (ConTim)

The following options are specified on the MASTER CONTROL CARD (**ConTim**). They can be listed in any order and can be omitted if the user wishes the option to take on its default value. Options may be continued onto several lines so long as a $ is used at the end of a continuation line. The command TYPE = CONTIM must be used on all MASTER CONTROL CARD (**ConTim**)s.

We first consider the "essential" commands; namely those commands which specify the model type and determine options which govern parameter restrictions, equation non-linearities, and likelihood weighting. Commands which govern which observations will be selected for the run are specified in the next set of options. Then options which determine the starting values, scaling, tolerances, and convergence criteria for iterative solution methods are listed. Then commands are given that control what print and punch output will be produced by the run. Finally detailed commands are given which are specific to each model type including dependent variables, time dependence parameter specifications, and heterogeneity options.

7.4.1. The STOP, MODEL, NONLIN, RESTRI, CONST, and WEIGHT Options

The following seven options are used to specify the type of model and the nature of the estimated equations and/or parameter restrictions. They will to a large extent determine the cost and complexity of a run. The model option determines which model type is used. The NONLIN option determines whether the model is linear or non-linear in the x's. The RESTRI command determines whether or not there are restrictions in the parameters. The WEIGHT and CONST commands determine whether observations are weighted and if a constant is used in equations. We now describe each command in detail indicating the possible options for each command.

Option	Description

TYPE = CONTIM
This command must always be used.

STOP This command instructs the program to stop. It should be placed at the end of the control file. The entire command STOP must be spelled out (no abbreviations).

MODEL = XX where XX is explained below. This option determines the model type used in the run. The models allowed in **ConTim** runs include the following (see the beginning of the Chapter for a more complete description).

- MODEL = MULTLO. The program assumes a multinomial logit model. Choice probabilities are given by a logistic choice model with x variables descriptive of the decision-maker -- hence constant across choices -- but where the parameter vector does vary by choice. This option cannot be used with NONLIN = 1.

- MODEL = CONDLO. The program assumes a conditional logit model. Choice probabilities are given by a logistic choice model with only one parameter vector, but where x variables vary by choice. This option cannot be used with NONLIN = 1.

- MODEL = FULLLO. The program assumes a mixed logit model. Choice probabilities are given by a full logistic choice model where variation of both the parameter vector and x variables across choices is set by the user. This is the only logistic choice model which can be used with NONLIN = 1. The NON-LINEAR EQUATION CARD is read, not the EQUATION VARIABLE CARD (even if NONLIN = 0).

- MODEL = ORDLO. The program assumes an ordered (N-Chotomous) logit model. Choice probabilities are given by a single x and parameter vector, where only the constant term varies by choice. Moreover, the choice constants are constrained to monotonically increase or decrease according to a preset ordering of the choices.

- MODEL = TPROB. The program assumes a tri-chotomous probit model. Precisely three unordered choices are assumed, and choice probabilities are assumed to be determined by a probit model with variation of the parameter vector and x vector across choices in a fashion set by the user. The NON-LINEAR EQUATION CARD is read, not the EQUATION VARIABLE CARD (even if NONLIN = 0).

- MODEL = NPROB. The program assumes an ordered (N-Chotomous) probit model. Choice probabilities are given by a single x and parameter vector, where only the threshold term varies by choice. Moreover, the choice thresholds are constrained to monotonically increase or decrease according to a preset ordering of the choices.

- MODEL = POISSON. The program assumes a Poisson regression model. The dependent variable is assumed to represent the number of occurrences of an event in a preset uniform time period. The estimated x'b are assumed to predict the log mean of the generating poisson process.

- MODEL = HAZARD. The program assumes a continuous time multi-state proportional hazard process. The dependent variable(s) represent beginning and ending "states" for time periods (spells) which can vary in length. The instantaneous transition probabilities are assumed to be exponential functions of the x variables. For 2-state problems this model form reduces to a simple failure time model.

- MODEL = EXP. The program assumes a discrete-time version of the proportional hazard model. The transition probabilities are given by the same exponential form as the continuous time version, but they describe the transition probability for a discrete time period rather than for an instantaneous point in time.

- MODEL = DBLEXP. The program assumes a discrete-time version of the proportional hazard model similar to MODEL = EXP. However, transition probabilities are given as a double exponential function of the x variables. Because of the exponential transformation used in the continuous time proportional hazard model, DBLEXP is the closest discrete time representation of that model, and without time dependence should yield almost equivalent parameters.

NONLIN = # where # is 0 or 1. The default is 0. This option determines whether linear or non-linear equation forms are used.

- If NONLIN = 0, equation(s) are linear in the x's (except for logit,

probit, exponential transforms) and are specified on the EQUATION VARIABLE CARD. This is the cheapest option, and should be used if model non-linearities are restricted to non-linearities in the parameters (see RESTRI). This option must be chosen if MODEL = CONDLO or MULTLO.

- If NONLIN = 1 then equations are non-linear in the x's and are specified on the NON-LINEAR EQUATION CARD(S). If NONLIN = 1 then starting values must be read in (START = 0). No starting values should be zero, however, as this will sometimes cause gradient problems. This option is the most expensive and should be used only when necessary. Although it does not have to be formally specified, this option is in effect required for MODEL = FULLLO or TPROB.

RESTRI = # where # is any non-negative integer. The default is 0. This option determines how many restrictions are imposed on parameters in estimation. Each restriction is specified separately on the RESTRICTION CARD(S). Any subset of the estimated parameters can be restricted to be a function of the unrestricted parameters, with estimates and standard errors appropriately adjusted. This option can be used with both linear and non-linear equations. If all system non-linearities can be expressed as non-linearities in the parameters, it is strongly advised that RESTRI, not NONLIN, be used. This allows the system to use the cheaper computation methods of NONLIN = 0. If restrictions are used, restricted coefficients will be marked with an "R" on printed output.

CONST = # where # is 0 or 1. The default is 0. This option determines if a constant is used in each model equation. If CONST = 0, it is. If CONST =1 it is not. This option can be overridden on an equation by equation basis by ading a constant as variable V0 (or CONSTANT) or taking one out with NV0 (or N[CONSTANT]). For MODEL = CONDLO the default option is no constant (CONST = 1), as a constant should not be included in all equations. CONST does not apply to the NON-LINEAR EQUATION CARD(S) where the constant must be specified manually (or included as a parameter).

WEIGHT = V#
where # is any integer from 1 to VAR*. This option determines if the sample log-likelihood is weighted. A variable name bracketed by [and] can also be used. If WEIGHT is not specified there is no weighting. The value in variable V# is used to weight each equation and time period (or spell) likelihood. Only positive weighting values are allowed. Thus all observations with zero or negative values for the weighting variable will be dropped. The weights will be scaled to average one so as not to effect the log-likelihood scale.

7.4.2. Options for Missing Data and the Inclusion of Observations in Estimation

The following options control how observations with missing data will be handled and what observations and/or time periods (relevant for panel data sets) will be used to estimate the equation(s).

Option	Description

MISS = # where # is 0, 1, or 2. The default is 0. This option determines the treatment of observations with missing data. It does not apply unless MISS is also specified on the DATA CONTROL CARD. If MISS = 0 (here) then no observations are excluded for missing data. If MISS = 1 then all variables used in the model (independent, dependent, and weight) are checked. If any have missing values the observation is excluded. If MISS = 2 (here) then any observation with any missing variables is excluded.

DISK = # where # is 0 or 1. The default is 0. This option determines internal disk usage for this model run only. It specifies a different option than DISK on the DATA CONTROL CARD.

- If DISK = 0 model run data is stored in core. This takes more core storage but is generally much cheaper in terms of CPU.
- If DISK = 1 the program will write model run data on the binary disk file INZ (set to 19) and not use as much core storage. If START = 1 and disk storage is used, then observations cannot be scaled (SCALE must = 1).

UTIME = $\#_1$ TO $\#_2$
where $\#_1$ is any integer from 1 to TIME and $\#_2$ is any integer from $\#_1$ to TIME. The default for $\#_1$ is 1 and for $\#_2$ is TIME (or the maximum number of time periods if variable time is used). This option determines the range of time periods used for the run.

XTIME = $\#_1$ TO $\#_2$
where $\#_1$ is any integer from 1 to TIME and $\#_2$ is any integer from $\#_1$ to TIME. This option determines the range of time periods which will not be used in the run. It cannot be used with UTIME.

UCROS = $\#_1$ TO $\#_2$
where $\#_1$ is any integer from 1 TO OBSV and $\#_2$ is any integer from $\#_1$ to OBSV. The default for $\#_1$ is 1 and for $\#_2$ is OBSV. This option determines the range of cross-sectional observations used for the run. Any units not in the specified range will not be used. With non-panel data it determines which observations are used.

XCROS = $\#_1$ TO $\#_2$
where $\#_1$ is any integer from 1 TO OBSV and $\#_2$ is any integer from $\#_1$ to OBSV. This option determines the range of cross-sectional observations which will not be used for the run. It cannot be used with UCROS.

UOBSV = $V\#_1$.XX.$V\#_2$ *or* UOBSV = $V\#_1$.XX.$\#_3$
where $\#_1$ and $\#_2$ are any integers from 1 to VAR*, $\#_3$ is any real number not expressed in scientific notation, and XX is one of the logical operators GT, LT, GE, LE, EQ, NE. There are no defaults. Variable names bracketed by [and] can be substituted for $V\#_1$ and/or $V\#_2$. If UOBSV is not specified, no observations will be excluded on this basis. This option specifies a FORTRAN IF

statement which determines which observations are used in a run. Any observations not meeting the criteria will not be used. $V\#_1$ specifies a variable whose value is compared either to the value in variable $V\#_2$ or the real number, $\#_3$ using the comparison indicated by the logical operator (conventional FORTRAN treatment, e.g., GT means greater than). If the comparison is true (i.e., V22.GE.3.27 or V27.EQ.V28 or [REEDER].NE.[NEW]) then the observation is used. The comparison is made separately for each time period of each cross-section unit.

7.4.3. Options for Starting Values and Choice of Numerical Optimization Methods

The following options control how starting values of parameters will be chosen and what sort of optimization techniques will be employed. There are also some options to control the tolerances which govern convergence of parameters and whether or not parameters should be scaled in estimation.

Option	Description
START = #	where # is 0, 1. The default is 0. This option determines the method of calculating coefficient starting values.

- If START = 0 starting values are read in on the COEFFICIENT STARTING VALUE CARD. This option must be used if NONLIN = 1 or MODEL = TPROB or FULLLO.

- If START = 1 the program uses regression methods to get starting values for the models in question. Depending on the type of model specification being estimated these starting values are computed in several different ways.

  - If MODEL = HAZARD or EXP or DBLEXP or POISSON or ORDLO or NPROB then starting values are determined by regressing the observed dependent variable against the independent variables for starting values. This is a 0/1 variable denoting a transition or not for HAZARD, EXP, and DBLEXP. For POISSON it is the number of occurrences, and for NPROB and ORDLO it is the number of the choice selected. The coefficients will then be scaled to reflect differences in the run model parameter partials and those implied for the regression used in the starting values. For example, with the MODEL = HAZARD, the model will adjust by a double exponentiation at the mean transition probability. With Box-Cox time duration, it is necessary to read in starting values for the exponents (LFIRST and LSECND).

  - If MODEL = MULTLO the program will run CHOICE-1 regressions of the second through CHOICE choices against the base group of choice one. These are binary linear probability regressions using in each case only those observations who selected choice 1 and the other choice included in the regression. Coefficients are adjusted for scalings.

- If MODEL = CONDLO the program runs a binary linear probability model regression stacking the chosen choice against all of the non-chosen ones. The coefficient vector is adjusted for scale.

FORMAT = # where # is 0 or 1. The default is 0. This option determines the format for all starting values.

- If FORMAT = 0 then all coefficients are read with (5D16.9) format (same as they are punched). No FORMAT CARDS are read.
- If FORMAT = 1 then the format for the coefficient vector is read preceding the STARTING VALUES CARD on the FORMAT CARD.

SCALE = # where # is 0 or 1. The default is 0. This option determines coefficient scaling.

- If SCALE = 0 then the coefficients are scaled to 1 by starting values. This is done for computational accuracy and does not effect the printout.
- If SCALE = 1 variables are not scaled. This option must be used if DISK = 1 on the MASTER CONTROL CARD (ConTim) and START = 1.

OUTLY = # where # is 0, 1, or 2. The default is 0. This option determines how the program handles outliers encountered during estimation.

- If OUTLY = 0 then the program will stop if it encounters an extreme observation in evaluating a probit/logit/exponential datapoint. This option can be used to avoid bad starting values.
- If OUTLY = 1 the program will set extreme outlier observations to boundary values if encountered, and will continue estimation. Any corrections will be printed.
- If OUTLY = 2 the program will make the same corrections as OUTLY = 1, but will not print the correction.

ESTIMA = # where # is 0, 1, 2, 3, 4, or 5. The default is 0. This

option determines the estimation iteration control. See the Appendix on Fletcher-Powell for more details.

- If ESTIMA = 0 the program uses both steepest descent and Fletcher-Powell iteration methods. The program will use the cross-products of the gradients after steepest descent as starting values for the Fletcher-Powell second derivatives matrix. If this is not invertible a partial identity matrix will be substituted.
- If ESTIMA = 1 the program stops after the steepest descent section.

- If ESTIMA = 2 same as option (=0) except that the program only uses Fletcher-Powell, no steepest descent. This option or (=3) should be used if time limits were reached on previous runs and a problem is being restarted in the middle.

- If ESTIMA = 3 same as option (=2) except that an identity matrix is used for starting values of the Fletcher-Powell second derivatives matrix not the cross-products of the gradients.

- If ESTIMA = 4 there will be no estimation at all. The program will compute coefficient standard errors using input values, and will perform residual analysis.

- If ESTIMA = 5 same as option (=4) except that coefficient standard errors are also not computed.

CNVRG = # where # is 0, 1, 2, 3, or 4. The default is 0. This option determines the criteria used for parameter convergence. See Appendix on Fletcher-Powell for more details.

- If CNVRG = 0 the program requires three criteria to all be satisfied for convergence. These are:

  1. The maximum element of the parameter gradient vector is less in absolute value than GTOL;

  2. The absolute proportional change in the function value is less than FTOL for two iterations (this criteria should not be used with TYPE = 3 with just-identified models); and

  3. The maximum absolute proportion change of any parameters is less than PTOL for two iterations.

- If CNVRG = 1, the program will stop when any of the convergence criteria specified in option (=0) are satisfied.

- If CNVRG = 2, the program will stop only when the gradient convergence criteria is satisfied.

- If CNVRG = 3, the program will stop only when the function change convergence criteria is satisfied.

- If CNVRG = 4, the program will stop only when the parameter change convergence criteria is satisfied.

GTOL = # where # is any real number not expressed in scientific notation > 0. The default is .00001. This option determines the tolerance used for convergence by the maximal absolute element of the parameter gradient vector.

FTOL = # where # is any real number not expressed in scientific notation > 0. The default is .0000001. This option determines the tolerance used for convergence by the absolute proportional change in the function value. It must be satisfied

for two iterations.

PTOL = # where # is any real number not expressed in scientific notation > 0. The default is .0001. This option determines the tolerance used for convergence by the maximal absolute proportional change of any element of the parameter vector. It must be satisfied for two iterations.

ITERA = # where # is any non-negative integer. The default is 50. This option determines the total combined number of iterations allowed in the steepest descent and the Fletcher-Powell section of program estimation.

SECS = # where # is any non-negative real number. The default is 5. This option determines the total number of CPU seconds allowed for both steepest descent and Fletcher-Powell sections of the program. This option applies only if a CPU timer is used in the program.

7.4.4. Options Controlling Output Printed and Output Punched

The following options control what information from estimation is printed out to the output and punch files.

Option	Description

PSTAN = # where # is 0, 1, 2, or 3. The default is 0. This option determines the method used to calculate standard errors of coefficients. Options (=1) (=2) or (=3) should be relevant only if multiple time periods are used.

- If PSTAN = 0 the program will use the inverse matrix of the negative of the sample log-likelihood 2nd derivative matrix evaluated at the estimated parameters to compute standard errors. This method will generally be correct for cross-sectional data, and by required assumption, for continuous time models as well.

- If PSTAN = 1 then the program corrects standard errors for intertemporal correlations (time-adjusted). If non-panel data is used this should have the same properties as option (=0). The parameter covariance matrix is given by D^{-1} S D^{-1} where D is the negative of the matrix of log-likelihood 2nd derivatives and S is a cross products matrix of 1st derivatives. Generally the 1st derivatives will be summed over all time periods for each cross-section before cross-producted. This option can be changed, however, if time-independence or a moving average time correlation specification is desired, by the option MAVER.

- If PSTAN = 2 then the program computes standard errors both ways by options (=0) and (=1).

- If PSTAN = 3 then the program computes standard errors from the inverse of the summed cross-products matrix of sample log-likelihood 1st derivatives evaluated for each cross-section (matrix S above). If estimates are maximum likelihood these standard errors will have

equivalent properties to those given by option (=0) or (=1). Generally with panel data the derivatives will be summed over time periods before they are cross-producted, but this option can be changed with option MAVER.

MAVER = # where # is any integer from 0 to TIME - 1. With panel data the default is TIME - 1. With pure time series or cross-sectional data the default is 0. This option determines the number of time period leads and lags used in summing log-likelihood first derivatives when the cross-products matrix of 1st derivatives is used in standard error calculations. This option applies only if TIME > 1 and PSTAN = 1 or 3. If MAVER = TIME - 1 then the program will sum derivatives for all time periods of each cross-section before cross-producting. If MAVER = 0 then derivatives will be cross-producted for each time period (generally correct only if equation errors are independent over time). If MAVER is between 0 and TIME - 1 then derivatives will be summed for MAVER time periods preceding and lagging each time period observation. This option is correct under the assumption that equation errors are correlated with a moving average process of order MAVER.

PCOVA = # where # is 0 or 1. The default is 0. This option determines if the full covariance matrix of estimated parameters is printed.

- If PCOVA = 0 the matrix is not printed.
- If PCOVA = 1 it is.

IPRINT = # where # is 0 or 1. The default is 0. This option determines if iteration information is printed. If IPRINT = 0, the program prints information on each iteration of the estimation process. If IPRINT = 1, then the program prints only summary information.

CWRITE = # where # is 0, 1, or 2. The default is 0. This option determines whether estimated parameters are punched (written on disk) on device IPC (generally set to 7).

- If CWRITE = 0 no coefficients are punched/written.
- If CWRITE = 1 the program punches/writes final coefficient (and inter-equation or time period correlations if ECORR > 0 or TCORR > 0) estimates on device IPC using format (5D16.9). Coefficient order is the same as printed.
- If CWRITE = 2 same as option (=1) except that coefficients (correlations) are also punched/written at each steepest descent/Fletcher-Powell iteration. This option is useful if the computer run can fail in the middle.

RPRINT = # where # is 0 or 1. The default is 0. This option determines whether the program performs and prints a residual/predicted value analysis.

- If RPRINT = 1, the program will compute predicted values, probabilities, expected value of the error for each observation/time

period analyzed. See RWRITE for details on what is printed.

- If RPRINT = 0 no residual analysis is done.

RWRITE = # where # is any non-negative integer. The default is 0. This option determines if residual analysis information is punched/written on disk for each observation/time period. This option can be used independently of RPRINT.

- If RWRITE = 0 nothing is punched.
- If # > 0 then # will be the device number that residual terms are punched/written on. Order of punch is the same as residual print, i.e., by time period, then cross-section observation.

For each time period/observation the following four items are punched (or printed for RPRINT)

7.4.5. Options that only Apply to the Logit/Probit Choice Models

The following options apply only if the MODEL run is one of the discrete choice model forms, MULTLO, CONDLO, FULLLO, ORDLO, TPROB, or NPROB. These models all have a similar model form. In each case the "dependent" variable is a "choice", depicted in **ConTim** by a discrete number 1,2, ... for a variable, DCHOICE. It will generally also be necessary to specify the number of choices available, CHOICE, or if the choice set size varies, the variable containing the choice set size, DSIZE. Several data options are also determined, CSCALE (for MULTLO), CSTAN (for TPROB), and TDATA (for CONDLO).

Option	Description

DCHOICE = V#
where # is any integer from 1 to VAR*. A variable name bracketed by [and] can be substituted for V#. This option determines the "dependent" variable of the model. The variable named in V# should contain the value, i.e. 1,2, ... , of the choice (or range value for ORDLO or NPROB) selected for each observation. The variable cannot have a value less than 1 or greater than CHOICE. This option is required for MULTLO, ORDLO, and NPROB. If it is not specified for CONDLO, FULLLO, or TPROB, it is assumed that data are adjusted so that choice 1 is always the chosen choice for each observation.

CHOICE = # where # is any integer greater than 1. The default is 2. This option determines the maximum number of choices in the choice set for any one observation. If the choice set size is constant for all observations (DSIZE is not specified) then CHOICE determines the choice set size. This option does not apply with TPROB, where the number of choices must be 3 (and is set at 3 by default). For MULTLO, CHOICE determines the number of estimated parameter vectors (CHOICE minus one).

DSIZE = V# where # is any integer from 1 to VAR*. A variable name bracketed by [and] can be substituted for V#. This option determines the variable which contains the choice set size if it varies by observation. The allowable values for a

choice set size must be between 2 and CHOICE. This option is allowed only for MULTLO, CONDLO, or FULLLO. For TPROB, ORDLO, and NPROB the choice set size must be constant. If the choice set size varies, it is assumed that choices are dropped from the end (thus for 4 choices 1,2,3,4 are available. and for 3 choices, 1,2,3).

TDATA = # where # is 0 or 1. The default is 0. This option only applies for CONDLO. It determines whether data is given for each choice.

- If TDATA = 0 the program assumes an x vector for each choice of the choice set for each observation. The actual data used by the program will be adjusted so that $x_i^* = x_i - x_j$ for the ith choice, where j is the selected choice for that observation. The program requires the specification of an x vector for all choices (CHOICE) on the EQUATION VARIABLE CARD.

- If TDATA = 1 the program assumes that the x vectors have already been adjusted to deviations about the values of the selected choice. Thus, the program reads specifications for only CHOICE-1 x vectors on the EQUATION VARIABLE CARD. A vector for choice 1 is not specified. Thus the option (=1) should be used only if the selected choice equals 1 for all observations (or data can be rearranged so that it is).

CSTAN = # where # is 0 or 1. The default is 0. This option only applies for TPROB. It determines the normalization used in estimating the covariance and variance terms of the choice error terms.

- If CSTAN = 0 the program normalizes the error variances of all three choices to one. The only estimated parameters will be the error correlation terms, σ_{12}, σ_{13}, and σ_{23}.

- If CSTAN = 1 the program does no normalization. The user must specify and estimate all six error covariance terms, the equation error sigmas and the three error correlations. To use this option parameter restrictions usually will have to be imposed.

CSCALE = # where # is 0 or 1. The default is 0. This option only applies for MULTLO. It determines how parameter vectors are normalized. Without further restrictions, the model fit will be the same regardless of which normalization is chosen.

- If CSCALE = 0 the program normalizes the multinomial logistic parameter vectors by setting the parameter vector of choice 1 to zero.

- If CSCALE = 1 the program normalizes the multinomial logistic parameter vectors by setting each element of the parameter vector of the the first choice equal to the negative of the sum of the corresponding elements of the second through CHOICE parameter vectors. This normalization implies that the parameter vectors will sum to zero.

7.4.6. Options for Poisson Regression Models

The following options apply only if the MODEL = POISSON option is chosen. They determine the "dependent" variable and repetition pattern.

Option	Description

DCOUNT = V#

where # is any integer from 1 to VAR*. A variable name bracketed by [and] can be substituted for V#. This option applies only for POISSON and determines the variable which contains the number of occurrences (the dependent variable) for each observation. There is no default. The variable V# can only take on non-negative integer values.

DSUMS = V# where # is any integer from 1 to VAR*. A variable name bracketed by [and] can be substituted for V#. This option applies only for POISSON and determines the variable which contains the number of repetitions for each observation. This is equivalent to weighting the observations, and should be used if the observations correspond to time periods (or wire lengths etc) of different lengths. V# should contain the number of repetitions for each observation. If DSUMS is not specified the program will assume that all observations have the same number of repetitions.

7.4.7. Options Applying Only to the Proportional Hazard Model

The following options apply only to the proportional hazard model, HAZARD, or its discrete counterparts, EXP or DBLEXP. The number of states (NSTATE) and different transition vectors (TRANS) are determined. Also specified is the data form (NDATA and FAIL) as well as the determination of which transitions occur (DFIRST, DSECND, DCOUNT, DBEFOR), and their timing (KNOWN). If duration dependence is assumed, it is set by DURA and LFIRST and LSECND. Finally, options are available for individual heterogeneity (HETERO, GAMMA and LAGUER).

Option	Description

NSTATE = # where # is a positive integer greater than one. The default is 2. This option determines the number of model "states" or the dimension of the transition matrix. If NSTATE = 2 then the model can represent the common failure time model.

TRANS = # where # is a positive integer. TRANS cannot be greater than NSTATE*(NSTATE-1). The default is 1. This option determines the number of different estimated transition probability functions. Each estimated transition vector can apply to as many different transitions as desired (see the TRANSITION VECTOR CARD). If a particular transition is not assigned to a transition vector on the TRANSITION VECTOR CARD its probability is assumed to be zero (any observation with this transition is dropped from the run).

DEP = $V\#_1$.XX.$V\#_2$ or DEP = $V\#_1$.XX.$\#_3$

where $\#_1$ and $\#_2$ are any integers from 1 to VAR*, $\#_3$ is any real number not expressed in scientific notation and XX is one of the logical operators (GT, LT,

GE, LE, EQ, NE). A variable name bracketed by [and] can be substituted for $V\#_1$ or $V\#_2$. This option only applies to the 2-state failure time model (NSTATE = 2, TRANS = 1, and transition is only allowed from state 1 to 2). The observation is assumed to fail when the comparison following DEP is true (using standard FORTRAN treatment of the operator). In this case the value in the variable $V\#_1$ is compared to another variable, $V\#_2$, or a real number, $\#_3$. The TRANSITION VECTOR CARDs are not read if DEP is used.

DFIRST = $V\#_1$ or DFIRST = $\#_2$

where $\#_1$ is any integer from 1 to VAR*, and $\#_2$ is any integer from 1 to NSTATE. The default is DFIRST = 1. A variable name bracketed by [and] can be substituted for $V\#_1$. This option determines the variable which contains the value of the state each observation is in at the start of its spell (In Con-Tim each observation is defined as a spell -- a period in which the x variables except for time dependence are constant, and at most one transition takes place). The variable specified by DFIRST can only take on the values from 1 to NSTATE. This option does not apply to the 2-state failure time model. If a number $\#_2$ is specified, observations will all be assumed to start in that state. This form is useful in estimating failure-time models with more than one type of failure. State 1 could be the starting state (no failure) for observations and different types of failure denoted by transitions into other states as indicated by the variable denoted by DSECND. This form can also be used to adopt the EXP or DBLEXP models to choice problems. All observations could "start" in state 1, and "transit" (choose) a choice indicated by DSECND. In this case, because of the nature of the probability model, choice 1 should be the "dominant" choice.

DSECND = V#

where # is any integer from 1 to VAR*. A variable name bracketed by [and] can be substituted for V#. This option determines the variable which specifies the state that each observation is in at the end of its spell. It does not apply to the 2-state failure model (see DEP). The variable specified by DSECND can only take on values from 1 to NSTATE.

DCOUNT = $V\#_1$ or DCOUNT = $\#_2$

where $\#_1$ is any integer from 1 to VAR* and $\#_2$ is any positive integer. A variable name bracketed by [and] can be substituted for $V\#_1$. This option determines how the length of each spell is specified. If DCOUNT is set equal to a positive integer, $\#_2$, then each spell is assumed to have a time length of $\#_2$. If $V\#_1$ is used, then the spell length is allowed to vary by observation, and is set by the value in the variable $V\#_1$. $V\#_1$ is allowed to only take on values from 1 to NSTATE in this case. If DCOUNT is not specified then all spells are assumed to have a length of one time period.

DBEFOR = V#

where # is any integer from 1 to VAR*. A variable name bracketed by [and] can be substituted for V#. This option is relevant only with time dependence (DURA > 0) and cannot be used with MODEL = EXP or DBLEXP. When used, the value in variable V# should be the number of time periods the observation was in its starting state (as set by DFIRST) before the start of each spell. Since **ConTim** defines a spell as a period of homogeneous x's and at

most one transition, it is possible for an observation to be in many spells without a transition because the x's are changing. It may be necessary to use DBEFOR in these instances to account for time spent in a state prior to the start of the sample for each individual. If DBEFOR is not used with duration dependent models, the program will count time from previous spells in adding up time in each state. In doing this it will take explicit account of cross-section units if the data file is panel or a pure time series, counting from period one for each individual, and assuming that the observation spells given for each individual represent continuous records. It will assume a zero previous time value for the first spell given for an individual.

NDATA = # where # is 0 or 1. The default is 0. This option determines if a separate x vector is used for each estimated transition parameter vector. It applies only if TRANS > 1.

- If NDATA = 0 the program uses a separate x vector for each estimated transition parameter vector. This option must be selected if NONLIN = 1. TRANS different x vectors must be read on the EQUATION VARIABLE CARD.

- If NDATA = 1 the program assumes a single x vector specified on the EQUATION VARIABLE CARD. The program will estimate different parameter coefficients for the x vector for each transition.

FAIL = # where # is 0 or 1. The default is 0. This option only applies to the 2-state failure time model (DEP = $V\#_1$.XX.$V\#_2$). It determines which observations are used for the run.

- If FAIL = 0 the program uses all acceptable observations.

- If FAIL = 1 and panel data is used, the program discards all observations for an individual following the spell (time period) where a failure (transition to state 2) occurs.

KNOWN = # where # is 0 or 1. The default is 0. This option only applies with MODEL = HAZARD. It can only be used currently with 2 state models as well. This option determines when the transition occurs within a spell.

- If KNOWN = 0 then the transition is assumed to occur (if one does) at the very end of the spell.

- If KNOWN = 1 then it is assumed that the precise timing of the transition is unknown. The program integrates over all possible transition points to determine the likelihood of each observation spell. In effect, the likelihood of a transition is solved for as one minus the probability of no transition occurring during the spell. This option is significantly more expensive than option (=0) and has a practical effect only when spell periods are of different length (see DCOUNT).

DURA = # where # is 0, 1, 2, 3, or 4. The default is 0. This option applies only for MODEL = HAZARD. It determines whether time (or duration) dependence is

assumed.

- If DURA = 0 the program assumes no time dependence. The hazard or transition probabilities do not depend upon the length of time an observation has been in a state.

- If DURA = 1 the program assumes a Weibull form of duration dependence. This is equivalent to adding a variable $\ell n(t) * \lambda$, where t is the length of time an observation has been in a state and λ is an estimated parameter, to the transition vector. It is also equivalent to multiplying the hazard, $e(x'\beta)$, by t raised to the λ power. A separate parameter λ will be estimated for each transition vector. This option, or option (=2) must be selected if x variables with time interactions are used.

- If DURA = 2 the program assumes a Gompertz form of duration dependence. This is equivalent to adding a variable $t * \lambda$, where t is the length of time an observation has been in a state and λ is an estimated parameter, to the transition vector. It is also equivalent to multiplying the hazard, $e(x'\beta)$, by $e(t*\lambda)$. A separate parameter λ will be estimated for each transition vector. This option, or option (=1) must be selected if x variables with time interactions are used.

- If DURA = 3 the program assumes a Box-Cox form of duration dependence. This is equivalent to adding the two terms, $((t ** \lambda_1 - 1) / \lambda_1) * \gamma_1$ and $((t ** \lambda_2 - 1) / \lambda_2) * \gamma_2$, to the transition vector. It is also equivalent to multiplying the hazard by e raised to the sum of theses two terms. For option (=3) the two parameters, γ_1 and γ_2, are free and estimated by the program separately for each transition vector. However, λ_1 and λ_2 are treated as fixed parameters and specified by the user with variables LFIRST and LSECND (if LFIRST and LSECND are not specified they default to 1 and 2). The values set for LFIRST and LSECND apply to all transition vectors. This is a very flexible functional form and can be used to represent many forms of duration dependence. For example if $\lambda_1 = 1$ and $\lambda_2 = 2$ it reduces to a quadratic in time.

- If DURA = 4 the program assumes a Box-Cox form of duration dependence as in option (=3). However, with this option all four parameters -- λ_1, λ_2, γ_1, and γ_2 -- are free and estimated for the program separately for each transition vector. LFIRST and LSECND are generally not read (unless regression starting values, START = 1, are used). Parameter restrictions (see RESTRI) can be used to restrict a subset of these parameters if the full Box-Cox specification is not desired, and option (=3) is not appropriate.

LFIRST = # where # is any real number not expressed in scientific notation. The default is 1. This option only applies if MODEL = HAZARD and DURA = 3 or 4. If DURA = 3 then LFIRST is the value of λ_1 used in the Box-Cox duration dependence formula for all transition vectors. If DURA = 4 and START = 1 then LFIRST is the starting value assumed for λ_1 (γ_1 is estimated by the regression).

LSECND = # where # is any real number not expressed in scientific notation. The default is 2. This option only applies if MODEL = HAZARD and DURA = 3 or 4. If DURA = 3 then LSECND is the value of λ_2 used in the Box-Cox duration dependence formula for all transition vectors. If DURA = 4 and START = 1 then LSECND is the starting value assumed for λ_2 (γ_2 is estimated by the regression).

HETERO = # where # is 0 or 1. The default is 0. This option only applies if MODEL = HAZARD and panel data is used. It determines if unobserved heterogeneity is assumed in the model.

- If HETERO = 0 the program assumes no heterogeneity.
- If HETERO = 1 the program assumes that there is an unobserved factor of heterogeneity (a random effect) which commonly effects all observations of a given individual. This random effect is assumed to be distributed with a gamma distribution, with a set shape parameter θ_1 which is supplied by the user (see GAMMA). The random effect is assumed to follow a one factor specification, with factor loadings for each transition vector estimated by the program. The heterogeneity parameters are estimated by integrating out the factor of heterogeneity using an approximate form of integration (see LAGUER).

GAMMA = # where # is any real number greater than zero not expressed in scientific notation. The default is 1. This option only applies if MODEL = HAZARD and HETERO = 1. It determines the value of the gamma distribution shape parameter, θ_1, used to integrate out heterogeneity.

LAGUER = # where # is 2-10, 12, or 15. The default is 10. This option only applies if MODEL = HAZARD and HETERO = 1. It determines the number of Laguerre integration points in the approximate numerical integration used to handle the gamma-distributed heterogeneity factor. The more Laguerre points used, the more accurate but expensive the approximation. Fifteen points are the maximum. [See Abramowitz and Stegun (1972, p.923]

7.5. EQUATION VARIABLE CARD(S)

These cards contains the specifications for the independent variables of the model transition probability or choice equations if NONLIN = 0. The number of cards read depends upon the following rules:

- No cards are read if NONLIN = 1 or MODEL = FULLLO or TPROB. In these cases, comparable but possibly non-linear, specifications are read on the NON-LINEAR EQUATION CARD.
- Precisely one EQUATION VARIABLE CARD is read (when NONLIN = 0) for MODEL = ORDLO, NPROB, POISSON, or MULTLO. The card specifies the x variables used in computing all choice probabilities (though parameters vary for MULTLO).

- If MODEL = CONDLO and TDATA = 0 then the EQUATION VARIABLE CARD is read CHOICE times. The equations define the x variables used for each choice. The number of variables shown on each card must be the same. Users are also cautioned not to use any variables which are constant across all choices for every observation, as these will cause the model to be unidentified. A constant, therefore, should generally not be included in the variable vector. If MODEL = CONDLO and TDATA = 1 only CHOICE-1 EQUATION VARIABLE CARDs are read, as that for choice 1 is excluded. Here as well, the same number of x's should be used in each equation.

- If MODEL = HAZARD or EXP or DBLEXP and NDATA = 1 then the program reads precisely one EQUATION VARIABLE CARD. The card lists the x variables used in computing all transition probability equations (though parameters will vary).

- If MODEL = HAZARD or EXP or DBLEXP and NDATA = 0 then the program reads a EQUATION VARIABLE CARD for each of TRANS transitions. Each card lists the x variables corresponding to each estimated transition equation. The estimated equations are allowed to contain different x's and to have a different number of parameters.

The EQUATION VARIABLE CARDs are used to specify the independent variables used in each transition or choice equation. The only exception is the constant which is automatically included if CONST = 0. If CONST = 1 (or MODEL = CONDLO) the constant may be selectively included as variable V0 or CONSTANT (it can't be included in strings). Variables are listed as $V\#_1 = \#_2$, where $\#_1$ is the number of the variable, and $\#_2$ is an optional code. Variable names can be substituted for $V\#_1$. Unlike most other cards, names do not have to be bracketed by [and] here (its optional).

Variable strings can also be used, indicated by a "-" between variables. Thus $V\#_1 - V\#_2 = \#_3$ (the $= \#_3$ is optional) implies that all variables from $\#_1$ to $\#_2$ will be included in the choice or transition equation (or given the code $\#_3$). If variables (including those in strings) are preceded by an "N", they will not be included. Thus, for example, V1-V10,NV8 implies that all variables from one to ten except variable eight will be included. If CONST = 0, the constant may be selectively excluded from the choice or transition equation by the use of NV0 or N[CONSTANT].

Variable names can be used in strings (the number is used to determine the string range) and with the N command. When names are used after N they must be bracketed by [and]. If a variable appears more than once on the EQUATION VARIABLE CARD its status will be determined by the last reference. Variables (or strings) must be separated by blanks or a comma.

Variables can be placed in any order on an EQUATION VARIABLE CARD. However, the program will always reorder them according to their variable numbers. Variable coefficients will be ordered in ascending order according to their numbers, followed by the constant (if used), for each choice or transition equation.

The variable code, $\#_2$, has three options.

- If $\#_2 = 0$ (the default if $= \#_2$ not included) the variable is included in the choice or transition equation. This option must be selected unless MODEL = HAZARD.

- If $\#_2$ = 1 the variable is included in the transition equation and is interacted with the Weibull (DURA = 1) or Gompertz (DURA = 2) time dependence variables. The variable added will be $V\#_1 * \ell n(t)$ (Weibull) or $V\#_1 * t$ (Gompertz). This option can be used only if MODEL = HAZARD and DURA = 1 or 2. A separate regular and interaction parameter will be estimated for the transition vector.

- If $\#_2$ = 2 the variable is not included in the transition equation but is included as an interaction term with time dependence terms as in option (=1). This option can be used only if MODEL = HAZARD and DURA = 1 or 2. Only the interaction parameter will be estimated.

7.6. NON-LINEAR EQUATION CARD(S)

If NONLIN = 1 specifications for the model's choice or transition probability equation(s) should be read on the NON-LINEAR EQUATION CARD(s). This form must also be used for all FULLLO or TPROB models. The NON-LINEAR EQUATION CARD forms are virtually identical to the EQUATION VARIABLE CARD except that freer, non-linear, equation specifications are allowed. Whereas only the x variables are listed on the EQUATION VARIABLE CARD(s) both the x's (V) and parameters (B) must be shown on the NON-LINEAR EQUATION CARD(s). The number of cards follows similar rules as the EQUATION VARIABLE CARD.

- If MODEL = FULLLO or TPROB then NON-LINEAR EQUATION CARDs are read for each choice (CHOICE times). The number of parameters and x's can vary across choices. However, care should be taken to make sure that the model is identified.

- If NONLIN = 1 and MODEL = ORDLO, NPROB, or POISSON then only one NON-LINEAR EQUATION CARD is read. If MODEL = ORDLO or NPROB choice thresholds (shifting constants) should not be specified, as they will be added by the program (the regular constant normally should be included).

- If MODEL = HAZARD or EXP or DBLEXP then the program reads a NON-LINEAR EQUATION CARD for each of TRANS transitions. Each card shows the equation of the corresponding transition probability vector.

Equation cards must have the following form. The first character is an equation delimiter, "E". The equation delimiter may be followed (optional) by a number indicating the choice or transition equation number (although equations must be read in order). This number need not be specified. The next character must be "=". After "=" the equation itself is spelled out. Each equation is made up of functions of variables (indicated by V# or names bracketed by [and]), parameters (indicated by B#), and numbers (#, can be real). The parameters can be numbered in any order, but all parameters from B1-B(max) must be specified on some equation card. If MODEL = ORDLO or NPROB then choice thresholds estimated as parameters will be assigned values after B(max) by the program (see the COEFFICIENT STARTING VALUE CARD). If MODEL = TPROB, the program will similarly assign estimated equation error sigmas (first) and correlations (second) after B(max) (see the COEFFICIENT STARTING VALUE CARD).

Each equation is made up of "elements" combined by the operations +, -, *, /. Each

"element" has the following form:

a * (or /) b ** c (i.e. ab^c)

where:

a = # (any real number not expressed in scientific notation), *and*

b = # (any real number not expressed in scientific notation), *or*

b = LN# (natural log of any positive number), *or*

b = V# (any variable. The name of the variable bracketed by [and] may be substituted for V#), *or*

b = LNV# (natural log of any variable. The name of the variable bracketed by [and] may be substituted for V#), *or*

b = B# (any parameter), *or*

b = LNB# (natural log of any parameter), *or*

b = (...) where the ... are other elements with a single layer of parenthesis, *or*

b = LN(...) natural log of what is in the parenthesis, *and*

c = # (any real number not expressed in scientific notation. There is a restriction however. If "c" is larger in absolute value than 1, it will be rounded to the nearest integer), *or*

c = V# (any variable. The name of the variable bracketed by [and] may be substituted for V#), *or*

c = B# (any parameter).

Note: only one layer of parenthesis is allowed and parenthesis cannot follow exponentiation (**). Any of the terms **a, b, c** are optional. Note as well, that sometimes it may be desired to normalize a choice equation as a "base group" for a logistic or probit choice system. This can be done by specifying E = 0 for the normalized choice.

Some examples are:

```
E=B1+B2*V2+B1*(B3**2+LNV6/B6)**V3
E2=B2**-B1-6.37+LN(B3/B1/B7*B2+3)*(B1+B2)
E=B1+[REEDER]**2*B2+([NEW]*B3+[REEDER]*V5*B4)**V2
E=B1*B2*([REEDER]+LN[NEW])+B2*[DEPTOB]
E=0
```

7.7. TRANSITION VECTOR CARD(S)

These cards are read only if MODEL = HAZARD or EXP or DBLEXP. The TRANSITION VECTOR CARD is also not read with 2-state failure time models (NSTATE = 2, TRANS = 1, DEP=V#.XX.V# OR #), as the transition (1,2) is assumed. The TRANSITION VECTOR CARD determines the state transitions whose probabilities are set by each estimated transition vector. As many transitions as desired can apply to each estimated parameter vector; however, at least one transition must apply. Any transition possibilities not linked to a transition vector are assumed to be forbidden. Observations showing one of these transitions will be excluded from the run. Note, that only transitions from one state to another are allowed.

One TRANSITION VECTOR CARD should be present for each (of TRANS) estimated transition equations. The card order will determine the meaning of the estimated parameter vectors. Each card should have the following format:

The first characters are the delimiters "TR". They can be followed (optionally) by a number indicating the transition vector's order. This is a check only and need not be specified. The next character must be "=". After "=" the applicable transitions should be listed, separated by commas or blanks. Each transition is a couplet, $(\#_1,\#_2)$. where $\#_1$ is the starting state and $\#_2$ is the ending state. These two numbers cannot equal one another, must be greater than zero, and cannot be larger than NSTATE. The (and) must be present, and the state numbers must be separated by a comma. A couplet can appear on only one TRANSITION VECTOR CARD. If more than one line is necessary for a card, use the $ continuation.

Some examples:

```
TR = (1,2)  (1,3)  (3,1)  (5,4)
TR2=(2,1) (3,2) (4,5) (5,1) (5,3)
TR3=(2,3)
TR=(5,2),(1,5)
```

7.8. RESTRICTION CARD(S)

Each restriction specified by the system must have a separate restriction card if RESTRI > 0. For each parameter restriction, a restriction equation is read in on a separate line. The restriction equations have the following form. On the left-hand side of the expression, one must specify the parameter or a function of the parameter to be restricted. In particular, the left-hand side of the restriction can take the following form:

a * (or /) b ** c = (i.e. ab^c)

where:

a = # (any real number not expressed in scientific notation), *and*

b = B# (the parameter to be restricted), *and*

c = # (any real number not expressed in scientific notation. There is a restriction however. If "c" is larger in absolute value than 1, it will be rounded to the nearest integer).

a and c are optional. **b** is not. Note that a parameter can be restricted only once.

Each restricted parameter (or function of a restricted parameter) can be restricted to be equal to functions of constants and/or unrestricted parameters. In particular the right-hand side of the restriction expression can take the following form:

a * (or /) b ** c (or -c) (i.e. ab^c)

where:

a = # (any real number not expressed in scientific notation), *and*

b = # (any real number not expressed in scientific notation), *or*

b = B# (any unrestricted parameter), *or*

b = (...) where the ... are other elements with a single layer of parenthesis, *and*

c = # (any real number not expressed in scientific notation. There is a restriction however. If "c" is larger in absolute value than 1, it will be rounded to the nearest integer).

Note: In the right-hand terms of the restriction, only one layer of parenthesis is allowed and parenthesis cannot follow exponentiation (**). Any of the terms **a, b,** c are optional.

In the specification of restrictions, the parameters are defined in the following way. If non-linear equations are used, then the parameters are specified using the same "B" numbers used to write out the equations on the NON-LINEAR EQUATION CARD(S). For MODEL = TPROB, error sigma and correlation parameters are numbered following the maximum parameter used in the non-linear equations. The numbering order is: The error sigma for choice (equation) one, the error sigma for choice two, the error sigma for choice three, the error correlation of choices one and two, the error correlation of choices one and three, and the error correlation of choices two and three. If program normalization is used (CSTAN = 0), then the choice sigmas are normalized to one, hence are not estimated or numbered or read in. For MODEL = ORDLO or NPROB, the threshold (or constant shift) parameters are numbered following the maximum parameter used in the non-linear equation (or regular equation). Recalling that the threshold break between choice one and two is normalized to 0, the first extra parameter is the threshold between choices two and three, the next between three and four etc. The last parameter is the threshold between choices CHOICE - 1 and CHOICE. These orderings assume that the first choice is that with the lowest y^*.

If linear systems are used (NONLIN = 0), all parameters are specified as B#, where the # denotes the order of the parameter. The parameter order is as follows. The coefficients corresponding to the variables in Choice or Transition equation One come *first*, with the coefficient number being equal to the ascending order of the variable numbers (i.e. V1,V2, ...) except for the constant (V0) which always is the last coefficient in each equation. If MODEL = HAZARD and DURA = 1 or 2 and time interactions are used, regular coefficients are fol-

lowed by interaction coefficients. Variables used in interactions are ordered by their numeric order regardless of the order they are read in. The regular time dependence parameter λ follows the interactions (if any). If MODEL = HAZARD and DURA = 3, regular coefficients are followed by the two free Box-Cox parameters, γ_1 and γ_2. If DURA = 4 all four Box-Cox parameters are ordered as, γ_1, γ_2, λ_1, and λ_2. This process is repeated for Equation Two etc. For MODEL = ORDLO or NPROB, threshold parameters follow as shown above.

Some examples of restriction cards are:

```
2.73/B2=B4/B5+B7**4
-B2**.5=B1*(B4+B5)**2+B6/B5**.5
```

7.9. COEFFICIENT STARTING VALUE FORMAT CARD

This card is read only if FORMAT=1 on the MASTER CONTROL CARD (ConTim) and START = 0. The FORTRAN format used to read coefficient values on the COEFFICIENT STARTING VALUE CARD is read here. It should be a standard FORTRAN format (with parenthesis) but not take more than one line. For example, (4F10.4,3X,3F12.3)

7.10. COEFFICIENT STARTING VALUE CARD

If START = 0, the parameter starting values should be read here by either the format of (5D16.9) if FORMAT = 0 or the format read on the COEFFICIENT STARTING VALUE FORMAT CARD. Parameter order is:

- For non-linear systems (NONLIN = 0 and EQUATION VARIABLE CARD used), the coefficients corresponding to the variables in Choice or Transition equation One come *first*, with the coefficient ordering being equal to the ascending order of the variable numbers (i.e. V1,V2, ...) except for the constant (V0) which always is the last coefficient in each equation. If MODEL = HAZARD and DURA = 1 or 2 and time interactions are used, regular coefficients are followed by interaction coefficients. Variables used in interactions are ordered by their numeric order regardless of the order they are read in. The regular time dependence parameter λ follows the interactions (if any). If MODEL = HAZARD and DURA = 3, regular coefficients are followed by the two free Box-Cox parameters, γ_1 and γ_2. If DURA = 4 all four Box-Cox parameters are ordered as, γ_1, γ_2, λ_1, and λ_2. This process is repeated for Equation Two etc. through all estimated choice or transition vectors. For MODEL = ORDLO or NPROB, the threshold (or constant shift) parameters are numbered following the maximum parameter used in the equation. Recalling that the threshold break between choice one and two is normalized to 0, the first extra parameter is the threshold between choices two and three, the next between three and four etc. The last parameter is the threshold between choices CHOICE - 1 and CHOICE. These orderings assume that the first choice is that with the lowest y^*.

- For non-linear systems (NONLIN = 1 or MODEL = FULLLO or TPROB), the parameters are ordered using the same "B" numbers used to write out the equations

on the NON-LINEAR EQUATION CARD(S). For MODEL = TPROB, error sigma and correlation parameters are numbered following the maximum parameter used in the non-linear equations. The numbering order is: The error sigma for choice (equation) one, the error sigma for choice two, the error sigma for choice three, the error correlation of choices one and two, the error correlation of choices one and three, and the error correlation of choices two and three. If program normalization is used (CSTAN = 0), then the choice sigmas are normalized to one, hence are not estimated or numbered or read in. For MODEL = ORDLO or NPROB, the threshold (or constant shift) parameters follow the maximum parameter used in the equation using the same ordering as in the linear case.

If NONLIN = 1 starting values equal to zero should generally be avoided (unless restricted) as they may cause problems.

7.11. TRANSITION CARD

This card signals the end of **ConTim** and the desire to read another data file. It has the following commands

Option	Description
SAMPLE	This command returns to the beginning of the program to read another DATA CONTROL CARD. If SUBROUTINES USER or USEO are used, the variable ISAMP will indicate the number of the sample being read. It can be accessed by including the statement: COMMON /G/ISAMP in the subroutine. The entire command SAMPLE must be spelled out (no abbreviations).
REWIND = #	where # is any positive integer. This option will rewind the file on device #. The entire command REWIND must be spelled out (no abbreviations).
ENDFILE = #	where # is any positive integer. This option will end file the file on device #. The entire command ENDFILE must be spelled out (no abbreviations).

7.12. SUMMARY OF ConTim OPTIONS

The following is an alphabetical list of the options available for the run section of ConTim, similar to that given for the regular **HotzTran** program in Chapter 3. We show the options of the MASTER CONTROL CARD (**ConTim**), as well as repeating those of the DATA CONTROL CARD for convenience. For each option, the following information is provided: (1) the card on which the option appears, (2) a brief description of the option, (3) the values the option can take, and (4) the page in the manual where the option is described in detail. The default value for each option is either underlined or noted as "(def. XXXX)."

Option	Card	Description of Option	Values	Page
			(0 generally means option not used)	
CHOICE	ConTim	The maximum number of choices (logistic or probit choice models)	2, > 2	127
CNVRG	ConTim	Convergence criteria control	0, 1, 2, 3, 4	124
CONST	ConTim	Constant control	0, 1	120
CSCALE	ConTim	MULTLO normalization control	0, 1	128
CSTAN	ConTim	TPROB normalization control	0, 1	128
CWRITE	ConTim	Coefficient punch/disk write control (0 = no punch)	0, 1, 2,	126
DBEFOR	ConTim	Variable number of previous spell time (HAZARD)	V#	130
DCHOICE	ConTim	Variable number of choice selected (logistic or probit choice models)	V#	127
DCOUNT	ConTim	Variable number for occurrences (POISSON)	V#	129
DCOUNT	ConTim	Variable number for spell length (HAZARD, EXP, DBLEXP)	V# or #	130
DEP	ConTim	Dependent variable number for two state failure time model	$V\#_1$.XX.$V\#_2$ or $\#_3$	129
DFIRST	ConTim	Starting state or variable number (HAZARD, EXP, DBLEXP)	V# or #	130

DSECND	ConTim	Ending state variable number (HAZARD, EXP, DBLEXP)	V#	130
DSIZE	ConTim	Variable number for choice set size (MULTLO, CONDLO, FULLLO)	V#	127
DSUMS	ConTim	Variable number for repetitions (POISSON)	V#	129
DISK	DATA	Internal observation disk control	0, 1, 2	22
DISK	ConTim	Internal model run data disk control	0, 1	121
DURA	ConTim	time dependence control (HAZARD)	0, 1, 2, 3, 4	131
ESTIMA	ConTim	Estimation procedure	0, 1, 2, 3, 4, 5,	123
FAIL	ConTim	two state failure model data control	0, 1	131
FORMAT	DATA	Data input format	0, (...), USER, USEO	21
FORMAT	ConTim	Format control for starting values	0, 1	123
FTOL	ConTim	Tolerance for function change convergence	> 0 (def. .0000001)	124
GAMMA	ConTim	Gamma shape parameter for HAZARD heterogeneity	1 > 0	133
GTOL	ConTim	Tolerance for gradient convergence	> 0 (def. .00001)	124
HETERO	ConTim	HAZARD heterogeneity control	0, 1	133
INPUT	DATA	Input device (FORTRAN) number	INP, > 0	22
IPRINT	ConTim	Iteration print control (1 = no print)	0, 1	126
ITERA	ConTim	Maximum number of iterations	0, >0 (def. 50)	125
KNOWN	ConTim	HAZARD transition timing control	0, 1	131
LAGUER	ConTim	Number of laguerre points (HAZARD heterogeneity)	2-10, 12, 15	133

LINE	DATA	Line width for printout (1 = 80 characters/line)	0, 1	22
LFIRST	ConTim	λ_1 value for DURA = 3	1. #	132
LSECND	ConTim	λ_2 value for DURA = 3	2. #	133
MAVER	ConTim	Number of moving average sums in std. error calculations	0 - TIME-1	126
MISS	DATA	Missing variable value code	<#, =#, ># (def. no miss)	23
MISS	ConTim	Equation missing value control	0, 1, 2	121
MODEL	ConTim	Model-type	ORDLO, NPROB, FULLLO, MULTLO, TPROB, CONDLO, POISSON, HAZARD, EXP, DBLEXP	118
NAME	DATA	Variable name control	0, 1	23
NDATA	ConTim	Separate x control for HAZARD	0, 1	131
NONLIN	ConTim	non-linear equation control	0, 1	119
NSTATE	ConTim	Number of states (HAZARD, EXP, DBLEXP)	2, > 2	129
OBSV	DATA	Number of cross-sectional units	0, > 0	20
OUTLY	ConTim	Outlier control	0, 1, 2	123
PCOVA	ConTim	Print control for parameter covariance	0, 1	126
PRINT	DATA	General data set print control	0, 1, 2, 3, 4, 5, 6, 7	22
PSTAN	ConTim	Standard error control	0, 1, 2, 3	125
PTOL	ConTim	Tolerance for parameter change convergence	> 0 (def. .0001)	125
RESTRI	ConTim	Number of parameter restrictions	0, > 0	120
RPRINT	ConTim	Residual analysis print control	0, 1	126
RWRITE	ConTim	Device number (FORTRAN) for	0, > 0	127

		residual punch		
SCALE	ConTim	Coefficient scaling control (1 = no scale)	0, 1	123
SECS	ConTim	Maximum number of CPU seconds	0, > 0 (def. 5)	125
START	ConTim	Starting value control	0, 1	122
STOP	ConTim	Termination command		118
TDATA	ConTim	CONDLO data form control	0, 1	128
TIME	DATA	Number of time periods or variable	1, > 1 or V#	21
TRANS	DATA	Number of data transformations	0, > 0	24
TRANS	ConTim	Number of estimated transition equations (HAZARD, EXP, DBLEXP)	1, > 1	129
UCROS	ConTim	Cross-sectional unit control	$\#_1$ TO $\#_2$ (def. all)	121
UOBSV	ConTim	Observation specific control (XX = .GT, etc.)	$V\#_1$.XX.$V\#_2$ or $\#_3$ (def. all)	121
UTIME	ConTim	Time period control (if TIME > 1)	$\#_1$ TO $\#_2$ (def. all)	121
VARI	DATA	Number of Variables read	> 0	20
WEIGHT	ConTim	Variable number for weighting	V#	120
XCROS	ConTim	Cross-sectional unit exclusion	$\#_1$ TO $\#_2$ (def. none)	121
XTIME	ConTim	Time period exclusion (if TIME > 1)	$\#_1$ TO $\#_2$ (def. none)	121

7.13. ConTim Examples

To illustrate the **ConTim** program in practice, we present a few sample runs illustrating its commands. Suppose that the same data set is used that is described on Chapter 5, consisting of 120 women observed for 5 time periods, with data on labor force participation, hours worked, wages, fertility, education, family income, age, kids, and race. In addition, suppose that we define two more variables: AGEYNG which is the age of the youngest child at the start of each year in the sample period and STATUS which is equal to 1 if the woman was

out of the labor force during year; equal to 2 if she was unemployed; and equal to 3 if she was working. These two variables follow the RACE variable in the data file. The first 10 lines in the data set, which consist of the data for the first individual for each of the five years and the data for the second individual for each of the five years, are repeated here:

```
1.0   500.  4.00  0.0   12.0  12000. 25.0 0.0   1.0   1.0   0.0   3.0
0.0   000.  0.00  0.0   12.0  12500. 26.0 0.0   1.0   1.0   0.0   2.0
1.0   2100. 4.50  0.0   12.0  13500. 27.0 0.0   1.0   1.0   0.0   3.0
0.0   000.  0.00  1.0   12.0  22000. 28.0 1.0   1.0   1.0   0.0   1.0
0.0   000.  0.00  0.0   12.0  20000. 29.0 1.0   1.0   1.0   1.0   2.0
0.0   000.  0.00  0.0   11.0  18000. 37.0 4.0   1.0   1.0   5.0   1.0
0.0   000.  0.00  0.0   11.0  20000. 38.0 4.0   1.0   1.0   6.0   1.0
0.0   000.  0.00  0.0   11.0  11500. 39.0 4.0   1.0   1.0   7.0   2.0
1.0   330.  3.10  0.0   11.0  12000. 40.0 4.0   1.0   1.0   8.0   3.0
1.0   1200. 3.90  0.0   11.0  18000. 41.0 4.0   1.0   1.0   9.0   3.0
                          .
                          .
                          .
```

where each line has a format of (12F6.0).

The following CONTROL CARDS specify a 2-state failure time model run, where units of time are measured in years, with the occurrence of a birth defined as "failure":

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(12F6.0) VARI=12
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE AGEYNG STATUS
TYPE=CONTIM DEP=V4 START=1 MODEL=HAZARD
HOURS WAGE EDUC INCOME AGE RACE
STOP
```

Suppose that we regard the different number of cumulative births (or achieved parity levels) as separate "states," i.e, state 1 = no births, state 2 = 1 birth, state 3 = 2 births, etc., and movement between different parity levels as transitions. Note that the only allowable transitions are from state 1 to state 2, from state 2 to state 3, etc. Moreover we let the hazard function for the transition to the first birth is different from the transition to the second birth but that the transitions to third and higher birth parities are the same; such a model is specified with the TRANSITION VECTOR CARDs. Assume that KIDS is the number of kids at the start of the period. Finally, assume that we want time dependence of a gompertz form, and use the same x's for all transitions.

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(11F6.0) VARI=11 TRANS=6
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE AGEYNG
V12=1
IF([KIDS].EQ.1) V12=2
IF([KIDS].EQ.2) V12=3
IF([KIDS].GE.3) V12=4
V13=V12
IF([FERT].EQ.1) V13=V13+1
STATE1 STATE2
TYPE=CONTIM START=1 MODEL=HAZARD DURA=2 TRANS=4 NDATA=1 $
DFIRST=[START1] DSECND=[START2] DBEFOR=[AGEYNG] NSTATE=5
HOURS WAGE EDUC INCOME AGE RACE
TR1=(1,2)
TR2=(2,3)
TR3=(3,4)
TR4=(4,5)
STOP
```

In this run. the "states" occupied at the beginning and end of each yearlong spell are obtained by creating the variables START1 and START2 just for this particular run. Note also that in the above specification DCOUNT is not explicitly specified since each "spell" is assumed to be one year long.

This data set can also be used to illustrate the choice models available in **ConTim**. Suppose that we wish to analyze the choice of the woman between the three labor market statuses: not in the labor force, unemployed and employed. This choice can be modeled using the STATUS variable as as a multinomial logit, or discrete EXP or DBLEXP choice as follows:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(12F6.0) VARI=12
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE AGEYNG STATUS
TYPE=CONTIM START=1 MODEL=MULTLO CHOICE=3 DCHOICE=[STATUS] $
CSCALE=0
FERT EDUC INCOME AGE KIDS HUS RACE
TYPE=CONTIM START=1 DFIRST=1 DSECND=[STATUS] NSTATE=3 TRANS=2 $
NDATA=1 MODEL=EXP
FERT EDUC INCOME AGE KIDS HUS RACE
TR=(1,2)
TR=(1,3)
STOP
```

This could be estimated as a probit model using the non-linear form as:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(12F6.0) VARI=12
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE AGEYNG STATUS
TYPE=CONTIM START=1 FORMAT=1 MODEL=TPROB DCHOICE=V11
E1=0
E2=B1*V4+B2*V5+B3*V6+B4*V7+B5*V8+B6*V9+B7*V10+B8
E2=B9*V4+B10*V5+B11*V6+B12*V7+B13*V8+B14*V9+ $
   B15*V10+B16
(16F2.0)
.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1
STOP
```

To examine whether the choices "out of labor force", "unemployed", and "working" are ranked, one could run:

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(12F6.0) VARI=12
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE AGEYNG STATUS
TYPE=CONTIM START=1 MODEL=ORLLO CHOICE=3 DCHOICE=[STATUS]
V4-V10
STOP
```

The Poisson regression model form is illustrated by the following model to determine the number of children each woman had at the end of the sample. Because the model looks at the accumulated total we run it as a pure cross-section (via the UTIME=(5TO5) command).

```
OBSV=120 TIME=5 INPUT=17 FORMAT=(12F6.0) VARI=12
       DATA ON FAMILY WORK FORCE DATA
LFP HOURS WAGE FERT EDUC INCOME AGE KIDS HUS RACE AGEYNG STATUS
TYPE=CONTIM START=1 MODEL=POISSON DCOUNT=[KIDS] UTIME=(5TO5)
V3 V5 V6 V7 V10
STOP
```

We illustrate the continuous time multi-state hazard specifications in which individuals have variable numbers of "spells" and in which these spells may be of varying length. Suppose that we have a data set obtained from a retrospective survey with "event histories" on births for 120 women from the time of their first marriage until the time of the survey. Suppose that the length of spells is measured in months. Recall that a spell is a consecutive period in which no x variables changed nor did the state being occupied. In this data set, we have the following variables:

FERT — A dummy variable equal to 1 if the woman gave birth to a child as of the end of a spell and 0 otherwise

EDUC — Number of years of education for the woman (constant over time)

INCOME — Annual Family income (in thousands of dollars), which is assumed to change each January and remaing constant for the subsequent 12 month period

AGE	The woman's age (in years) as of January
KIDS	The number of children the woman had given birth to as of the beginning of the current spell
HUS	A time-varying dummy variable equal to 1 if the woman had a husband during the current spell and 0 otherwise
RACE	A dummy variable equal to 1 if the family was white (constant over time)
SPELL#	A variable equal to the number of the current spell for a particular woman
LENGTH	A variable equal to the length of the current spell (in months)

Consider the data for two women in this data file. The *first woman* is married in June, has her first birth 26 months later and gets divorced after 45 months of marriage. Finally, suppose that the interview occurred 5 months after her divorce. In addition, her INCOME and AGE variables change on January of each year. The *second woman* is married in December, has her first child 12 months after marriage and has her second child 20 months after the second. She is interviewed 2 months after the birth of her second child. Again, her INCOME and AGE variables change on January of each year. Based on these event histories, the first woman has 7 spells and the second has 6 spells; that is the first woman as 7 lines of data and the second has 6. Their data looks like the following:

```
0.0   12.0  12.0  20.0  0.0   1.0   1.0   -1.   7.0
0.0   12.0  20.5  21.0  0.0   1.0   1.0   2.0   12.0
1.0   12.0  16.3  22.0  0.0   1.0   1.0   3.0   7.0
0.0   12.0  16.3  22.0  1.0   1.0   1.0   4.0   5.0
0.0   12.0  25.3  23.0  1.0   1.0   1.0   5.0   12.0
0.0   12.0  27.2  24.0  1.0   1.0   1.0   6.0   3.0
0.0   12.0  27.2  24.0  1.0   0.0   1.0   7.0   5.0
0.0    9.0  18.0  18.0  0.0   1.0   0.0   -1.0  1.0
1.0    9.0  19.5  19.0  0.0   1.0   0.0   2.0   11.0
0.0    9.0  19.5  19.0  1.0   1.0   0.0   3.0   1.0
0.0    9.0  17.8  20.0  1.0   1.0   0.0   4.0   12.0
1.0    9.0  18.1  21.0  1.0   1.0   0.0   5.0   7.0
0.0    9.0  18.1  21.0  2.0   1.0   0.0   6.0   2.0
                         .
                         .
                         .
```

Note that the variable SPELL# = −1.0 at the beginning of the data for a new individual as described in the TIME option discussed on page 21.

Again, we regard the different number of cumulative births (or achieved parity levels) as separate "states," i.e, state 1 = no births, state 2 = 1 birth, state 3 = 2 births, etc., and movement between different parity levels as transitions and the only allowable transitions are from state 1 to state 2, from state 2 to state 3, etc. The model we are interested in fitting assumes that each birth transition has a separate set of coefficients, i.e., the birth of the first child has a different hazard function than the birth of the second, and the second as different that the third etc.; such a model is specified with the TRANSITION VECTOR CARDs. Such a model

can be estimated with the following CONTROL CARDs:

```
OBSV=120 TIME=V8 TIME=18 INPUT=17 FORMAT=(9F6.0) VARI=9 TRANS=8
      CONTINUOUS TIME BIRTH PROCESS MODEL
FERT EDUC INCOME AGE KIDS HUS RACE SPELL# LENGTH
V10=1
IF([KIDS].EQ.1) V10=2
IF([KIDS].EQ.2) V10=3
IF([KIDS].EQ.3) V10=4
IF([KIDS].EQ.4) V10=5
IF([KIDS].EQ.5) V10=6
V11=V10
IF([FERT].EQ.1) V11=V11+1
STATE1 STATE2
TYPE=CONTIM START=1 MODEL=HAZARD DURA=2 TRANS=6 NDATA=1 $
DFIRST=[STATE1] DSECND=[STATE2] DCOUNT=[LENGTH] NSTATE=7
EDUC INCOME AGE RACE HUS
TR1=(1,2)
TR2=(2,3)
TR3=(3,4)
TR4=(4,5)
TR5=(5,6)
TR6=(6,7)
STOP
```

where the maximum number of spells for a single individual in the sample is 18.

Finally, we illustrate a conditional logit run. Suppose the 120 women represent a cross-section of women who just took new jobs. Moreover suppose that the women had a choice of up to 4 jobs, and the variables WAGE, DISTANT, and START each with a subscript of 1 to 4 represent the wages, travel distance, and starting hour of the up to four jobs each woman considered. (This data set is not illustrated herein.) Let JOBS represent the number of choices and SELECT be the number of the job actually selected by the woman. Note that the jobs can be ordered in any way, except that if fewer than 4 jobs were considered the variables WAGE1, DISTANT1, START1, must be filled in first. To estimate the weights attached to the three variables in making job choices, we run:

```
OBSV=120 TIME=1 INPUT=17 FORMAT(15F6.0) VARI=15
       DATA ON JOB SELECTION
JOBS SELECT AGE WAGE1 DISTANT1 START1 WAGE2 DISTANT2 START2
WAGE3 DISTANT3 START3 WAGE4 DISTANT4 START4
TYPE=CONTIM START=1 MODEL=CONDLO CHOICE=4 DCHOICE=[SELECT] $
    DSIZE=[JOBS] TDATA=0
WAGE1 DISTANT1 START1
WAGE2 DISTANT2 START2
WAGE3 DISTANT3 START3
WAGE4 DISTANT4 START4
STOP
```

The conditional logit model can be used to see if younger women are more wage sensitive, we create a variable which interacts age with wage, A*WAGE.

```
OBSV=120 TIME=1 INPUT=17 FORMAT(15F6.0) VARI=15 TRANS=4
       DATA ON JOB SELECTION
JOBS SELECT AGE WAGE1 DISTANT1 START1 WAGE2 DISTANT2 START2
WAGE3 DISTANT3 START3 WAGE4 DISTANT4 START4
[A*WAGE1]=V3*V4
[A*WAGE2]=V3*V7
[A*WAGE3]=V3*V10
[A*WAGE4]=V4*V13
TYPE=CONTIM START=1 MODEL=CONDLO CHOICE=4 DCHOICE=[SELECT] $
    DSIZE=[JOBS] TDATA=0
WAGE1 DISTANT1 START1 A*WAGE1
WAGE2 DISTANT2 START2 A*WAGE2
WAGE3 DISTANT3 START3 A*WAGE3
WAGE4 DISTANT4 START4 A*WAGE4
STOP
```

8. REFERENCES

Abramowitz, M. and I. Stegun, eds. (1972), Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, National Bureau of Standards, Applied Mathematics Series No. 55, revised December 1972.

Altman, E., R. Avery, R. Eisenbeis, and J. Sinkey, (1981) Application of Classification Techniques in Business, Banking, and Finance, Greenwich Ct: JAI Press, Inc.

Amemiya, T. (1974), "The Nonlinear Two-Stage Least Squares Estimator," Journal of Econometrics, Vol. 2, 1974, 105-110.

Amemiya, T. (1977), "The Maximum Likelihood and Nonlinear Three-Stage Least Squares Estimator in the General Nonlinear Simultaneous Equations Model," Econometrica, Vol. 45, 1977, 995-968.

Avery, R., L. P. Hansen, and V. J. Hotz (1983), "Multiperiod Probit Models and Orthogonality Condition Estimation," International Economic Review, vol. 24, No. 1, February 1983, pp. 21-35.

Avery, R. and V. J. Hotz (1984), "Statistical Models for Analyzing Absentee Behavior," in Absenteeism R. Atkin and P. Goodman eds., San Francisco: Jossey-Bass Inc., 158-193.

Avery, R. and V. J. Hotz (1984), "Estimating Systems of Equations with Limited Dependent Variables," unpublished manuscript, NORC, October 1984.

Butler, J.S. and R. Moffitt (1982), "A Computationally Efficient Quadrature Procedure for the One-Factor Multinomial Probit Model," Econometrica, Vol. 50, May 1982, pp. 761-764.

Chamberlain, G. (1982), "Panel Data," Discussion Paper No. 8209, SRI, April 1982.

Fletcher, R. and M. J. D. Powell (1963), "A Rapidly Convergent Descent Method for Minimization," The Computer Journal 6, pp. 163-168.

Flinn, C. and J. Heckman (1982), "Models for the Analysis of Labor Force Dynamics," in Advances in Econometrics, R. Bassman and G. Rhodes, eds., Vol. 1, Greenwich Ct.: JAI Press, Inc, pp. 35-95.

Flinn, C. and J. Heckman (1983a), "Are Unemployment and out of the Labor Force Behaviorally Distinct Labor Force States?" Journal of Labor Economics Vol. 1, 1983, pp. 28-42.

Flinn, C. and J. Heckman (1983b), "The Likelihood Function for the Multistate-Multiepisode Model in 'Models for the Analysis of Labor Force Dynamics'" in advances in Econometrics, R. Bassman and G. Rhodes, eds., Vol. 2, Greenwich Ct.: JAI Press, Inc.

Gallant, A. R. and D. Jorgenson (1980), "Statistical Inference for a System of Simultaneous, Nonlinear Implicit Equations in the Context of Instrumental Variable Estimation," Journal of Econometrics, Vol. 11, 1979, 275-302.

Griliches, Z., B. Hall and J. Hausman (1978), "Missing Data and Self-Selection in Large Panels," Annales de l'insee, April-September 1978, pp. 137-176.

Gruvaeus, G. T. and K. G. Jöreskog (1970), "A Computer Program for Minimizing a Function of Several Variables", Educational Testing Service Research Bulletin RB-70-14, Princeton, New Jersey.

Hansen, L. P. (1982), "Large Sample Properties of Generalized Method of Moments Estimators," Econometrica, Vol. 50 No. 4, July 1982, pp. 1029-1055.

Hauser, R., S.-L. Tsai, and W. Sewell (1981), "A Model of Stratification with Response Error in Social and Psychological Variables," Unpublished Manuscript, University of Wisconsin, October 1981.

Heckman, J. (1978), "Dummy Endogenous Variables in a Simultaneous Equations System," Econometrica, Vol. 46, 1978, pp. 931-959.

Heckman, J. (1981), "Statistical Models for Discrete Panel Data," in Structural Analysis of Discrete Data with Econometric Applications, C. Manski and D. Mcfadden, eds., Cambridge, MA: MIT Press, pp. 114-177.

Heckman, J. and T. MaCurdy (1980), "A Life Cycle Model of Female Labour Supply," Review of Economic Studies, Vol. 47, 1980, pp. 47-74.

Heckman, J. and B. Singer (1982), "The Identification Problem in Econometric Models for Duration Data," in Advances in Econometrics, W. Hildenbrand ed., London: Cambridge University Press, pp. 39-77.

Hotz, V. J. (1983), "A Structural Model Life Cycle Fertility and Female Labor Supply," Unpublished Manuscript, Carnegie-Mellon University, May 1983.

Hotz, V. J., F. Kydland, and G. Sedlacek (1982), "Intertemporal Substitution and Labor Supply," Unpublished Manuscript, Carnegie-Mellon University, December 1982.

Jöreskog, K. (1973), "A General Method for Estimating a Linear Structural Equation System," in Structural Equation Models in the Social Sciences, A. Goldberger and O. Duncan, eds., New York: Seminar Press, 1973.

Jöreskog, K. and A. Goldberger (1975), "Estimation of a Model with Multiple Causes of a Single Latent Variable," Journal of the American Statistical Association, Vol. 70, 1975, 631-639.

Jöreskog, K. and D. Sörbom (1977), "Statistical Models and Methods for Analysis of Longitudinal Data," in Latent Variables in Socioeconomic Models, D. Aigner and A. Goldberger, eds., New York: North-Holland Publishing Company, 1977.

Jorgenson, D. and J.F. Laffont (1974), "Efficient Estimation of Nonlinear Simultaneous Equations with additive Disturbances," Annuals of Economic and Social Measurement, 3, pp. 615-640.

Kalbfleisch, J. and R. Prentice (1980), The Statistical Analysis of Failure Time Data, New York: John Wiley.

MaCurdy, T. (1981), "An Intertemporal Analysis of Taxation and Work Disincentives," Unpublished Manuscript, Stanford University, December 1981.

Maddala, G. S. (1978), "Selectivity Problems in Longitudinal Data," Annales de l'insee, April-September 1978, pp. 137-176.

Maddala, G. S. (1983), Limited-Dependent and Qualitative Variables in Econometrics, Cambridge: Cambridge University Press.

McFadden D. (1973), "Conditional Logit Analysis of Qualitative Choice Behavior," in Frontiers in Econometrics, P. Zarembka, ed., New York: Academic Press.

McFadden D. (1981), "Econometric Models of Probabilistic Choice," in Structural Analysis of Discrete Data with Econometric Applications, C. Manski and D. McFadden, eds., Cambridge, Mass.: M.I.T. Press.

Nerlove, M. and J. Press (1973), "Univariate and Multivariate Log-linear and Logistic Models," RAND report R-1306-EDA/NIH.

White, H. (1982), "Instrumental Variables Regression with Independent Observations," Econometrica, Vol. 50, 1982, pp. 817-838.

Winship, C. and R. Mare (1981), "Structural Equations and Path Analysis for Discrete Data," American Journal of Sociology, forthcoming.

Appendix I
PROGRAM ERROR MESSAGES

The program will print out error messages if it detects them in processing control cards. The error message will generally be printed after the "cards" which caused the problem, and will ordinarily cause termination of the program. The program will print out all input cards labeled by the name of the card that the program thinks it is reading. The program will only be able to catch errors which unambiguously violate program specifications and thus may fail to catch some potentially troublesome problems. Some guides to interpreting the relatively terse error messages are as follows.

ERROR 1 "READ EOF. ON CONTROL CARD". This message means that the end of the card file was reached before the program expected it. The likely cause is a line missing or out of place. A clue to this cause is a card which is printed with an incorrect name. Another cause could be the failure to use a $ continuation break when you should have, specifying too many equations, variables etc. or including too few variable names etc.

ERROR 2 "INVALID OPTION CARD". This option means that the program could not translate a command (such as VARI=2). Whenever this error occurs, the program will print out the approximate code that it was processing when it detected the error. The likely causes are

- Failure to leave a space or comma between commands.
- Failure to spell a command correctly.
- Leaving out the '=' sign in a command.
- Getting a card out of order so a command comes in the wrong place.
- Failure to put [and] around a variable name when used in a command.
- Forgetting parenthesis in a FORMAT or IF statement.
- Placing a $ break in the middle of a command.
- Placing blanks in the middle of a command.

ERROR 3 "INCONSISTENT OPTIONS REQ". This message means that mutually exclusive options were requested. The meaning depends on which "card" contains the error.

- If on the DATA CONTROL CARD possible causes are:
  - VARI is not specified;
  - DISK = 1 and FORMAT or TRANS is not 0;
  - OBSV = 0, FORMAT = 0, and INPUT is the same file as the

control card file;

- TIME = 0 or is given a variable number out of range or a variable is named instead of numbered.
- TIME is determined by a variable, but the maximum number of time periods is not specified by a separate TIME = # command.

- If on the MASTER CONTROL CARD possible causes are:
  - for UTIME, XTIME, UCROS, or XCROS the 1st specified number is larger than the 2nd or numbers are out of range for the sample;
  - TYPE is not specified;
  - TYPE is not equal to 3 and TEXOG is greater than 0 or START = 1;
  - TEXOG = 0 and TSUB is greater than 0;
  - TYPE = 1 and START is greater than 2;
  - TYPE = 3 and TEXOG is greater than 0 and TLAG is greater than LAGS;
  - TYPE or MODEL is out-of-range or option is misspelled;
  - ECORR or TCORR is greater than 0, TYPE = 2 or 4, or EQUA = 1 (ECORR) or TIME = 1 (TCORR);
  - NONLIN = 1 and START = 2 or 4;
  - TYPE = 2 and HERMIT is out of range, RANDOM = 2 and EQUA = 0 or 1, RANDOM = 0 or 1 and TIME = 1;
  - TYPE = 4 and EQUA = 1.
- If on the BASIC EQUATION CARD possible causes are:
  - TYPE = 2 or 4 and MODEL = LOGIT or TRUNCATE;
  - FIXED = 1 and, TIME = 1 or TYPE = 2 or TYPE = 4.
- If on the EQUATION TESTING VARIABLE CARD it means that orthogonality conditions were imposed in the basic model which are not imposed in the test.
- If on the MASTER CONTROL CARD (MINREG) possible causes are:
  - for UTIME, XTIME, UCROS, or XCROS the 1st specified

number is larger than the 2nd or numbers are out of range for the sample;

- TYPE = MINREG is not specified on the first card;
- No dependent and/or independent variable cards;
- WEIGHT, RPRINT, RWRITE, AUTO, or MISS used with DISK = 3 OR 5 on the DATA CONTROL CARD.

- If on the MASTER CONTROL CARD (CONTIM) possible causes are:
  - for UTIME, XTIME, UCROS, or XCROS the 1st specified number is larger than the 2nd or numbers are out of range for the sample;
  - TYPE = CONTIM is not specified, or an incorrect MODEL form;
  - More transition vectors specified than allowed, or NSTATE < 2;
  - KNOWN used with NSTATE greater than 2;
  - A dependent variable is not specified (except CONDLO);
  - TPROB is used and NSTATE is not 3;
  - NONLIN = 1 and MODEL = MULTLO or CONDLO or START = 1;
  - DURA, NFUN, OR KNOWN used and MODEL not equal to HAZARD;
  - DEP used with more than 2 states or 1 transition vector.

ERROR 4 "VARIABLE # OUT OF RANGE". A variable number specified is less than 0 or greater than VAR*. This can occur because of a typing mistake or a card out of order. It can also occur if a variable name is not bracketed by [and] where it should be or the characters = or - or $ or STOP or SAMPLE are used in names not bracketed. If the error occurs in specifying the dependent variable in an equation it can also mean that .EQ. or .NE. were used with TOBIT or DBLTOB, or that .LE., .LT., .GE., .GT. were not specified for DBLTOB.

ERROR 5 "CORE EXCEEDED DURING READ". This error can occur when OBSV = 0 and a read to end-of-file is used. It means that more internal core storage is needed and was detected at the observation listed.

ERROR 6 "READ ERROR ENCOUNTERED". This message implies that a read error was encountered in the observation file at the observation listed. If this error occurs at observation 0 it means that the file could not be accessed at all (FORMAT

is wrong, file doesn't exist etc.) This error can also occur when the observation time period is set by an input variable (TIME = V3) and a time period out of the acceptable range is encountered.

ERROR 7 "RESTR/TRANS/NON-LINEAR CARD". This message implies that the program had trouble decoding a restriction, transformation, or non-linear equation card. The program will generally printout the approximate area of code that it was processing when it detected the error. Its causes will depend on the place of the error.

- General causes on any card are:
  - LN or EXP misspelled;
  - two layers or unmatched parenthesis or parenthesis after **;
  - no '=' sign specified;
  - failure to separate equation elements by - + * / or typing two operators in a row (++);
  - variables or parameters out of range;
  - variable names used without brackets [and];
  - the use of any unallowed characters.
- If the error occurs on a TRANSFORMATION CARD possible additional causes are:
  - FOR, IF, IFTIME or IFCROS misspelled or parenthesis not used;
  - the first element of the range of the FOR, IFTIME, or IFCROS statement is more than the second or is not a positive integer;
  - The index variable "I" following a FOR statement is not properly bracketed by < and >;
  - a variable is not specified on the left-hand-side of the equation or is out of range;
  - a variable on the right-hand-side of the equation or in an IF statement is created after the current transformed variable (or not at all);
  - leads or lags of variables more than the total number of time periods;
  - LN or EXP used after **.
- If the error occurs on a NON-LINEAR EQUATION CARD possible

additional errors are:

- E, S, V, P, or C are not on the left-hand-side of the equation or equations are out of order or left out;
- parameter specifications are not complete (some numbers in the middle are not specified);
- EXP or variable leads/lags are used.

- If the error occurs on a RESTRICTION CARD possible additional errors are:
  - V, LN, EXP, or leads/lags are used;
  - The left-hand-side of the equation does not contain a parameter specification;
  - a restricted parameter appears on the right-hand side of another equation;
  - a parameter appears after exponentiation **.

- If the error occurs on a TRANSITION VECTOR CARD in CONTIM the error indicates that a transition is out of range either with too large a state specified or too small. It may occur because of a failure to include a = , or (.

ERROR 8 "MORE WORDS OF CORE NEEDED". This message implies that more core storage is needed. The amount needed is listed. Possible remedies are to provide more core (see PROGRAM CHANGES FOR SYSTEM ADAPTION) or to reduce core needs by putting observations on disk (see the command(s) DISK). This error may be symptomatic of other errors such as variables out-of-range, OBSV being misspecified etc.

ERROR 9 "WEIGHTING MATRIX". This message indicates that the program was unable to invert the "optimal weighting" matrix formed if TYPE = 3 and OPT = 0. Since this matrix is formed as the cross-products matrix of the orthogonality conditions, failure to invert indicates that the orthogonality conditions are not linearly independent. This can occur if there are too few observations, if variables are constant over time and not so specified, or if orthogonality conditions are redundant. This error causes termination of the model run but not the job.

ERROR 10 "COMPUTE STARTING VAL.". This message indicates that the program was unable to compute starting values for the model run from the preliminary regression. Non-identification of the parameters is a likely cause. This error causes termination of the model run but not the job.

ERROR 11 "COMPUTE STANDARD ERR.". This message indicates that the program could not properly compute standard errors. (thus those printed should not be trusted). A hessian matrix not of full rank is the likely cause perhaps from parameter non-identification. If panel data is used this error can occur even

when parameters are identified when the number of cross-section units is less than the number of parameters. This error causes termination of the model run but not the job.

ERROR 12,13 "STEEPEST DESCENT" or "FLETCHER-POWELL". These errors indicate that the program could not calculate function values and gradients. The most likely cause is an observation value way out of range caused either by

- a lack of proper scaling of data;
- bad starting values; or
- a misspecified model (particularly non-linear); or
- poor tolerance with a fixed effect.

The program will generally printout the number and time period of the specific observation which caused the trouble. The numbers refer to the number within the specific range used for the run.

ERROR 14 "IN VARIABLE NAME". This error occurs when the program attempts to match a variable name to its number. It can occur in processing names on the DATA TRANSFORMATION, MASTER CONTROL, and the BASIC EQUATION CARDS where variable names must be bracketed. It can also occur on the TIME VARIATION, EQUATION VARIABLE, and EQUATION TESTING VARIABLE CARDS where they do not have to be bracketed. The program will printout the name of the variable it was processing when the error was detected. Likely causes are:

- misspelling a name or using an incorrect name;
- using a variable name twice;
- forgetting the =C on the TIME VARIATION CARD;
- using the characters = or - or $ or STOP or V# or NV# in a name;
- not placing a comma or space between variable names;
- placing an inappropriate character after the = sign on the EQUATION VARIABLE CARD.

ERROR 15 "MASTER CARD ORDER". This error occurs when the program expects to be reading a MASTER CONTROL CARD with the specification TYPE = 1, 2, 3, 4, MINREG or CONTIM on it, but encounters a different card. The most likely cause is a card out of order or missing.

The program will also print out an error message when outliers are encountered in evaluating probit/logit/Tobit/double Tobits/truncated observations and OUTLY = 0 or 1. The program will print out the observation/time period/equation number where the trouble was encountered. The program will also printout an error message if a parameter such as a variance

or correlation has an inadmissible value

Appendix II
STEEPEST DESCENT AND FLETCHER-POWELL MINIMIZATION PROCEDURES

The Steepest Descent and Fletcher-Powell procedures can be used to iteratively solve for the minimum of any function given analytic 1^{st} derivatives and, if desired, analytic 2^{nd} derivatives. The methods as used in our program can be described as follows.

Let F_i be the value of the function to be minimized evaluated at K parameter values b_i. Let $g_i = \partial F_i/\partial b_i$ be the vector gradient of the function. Finally, let $G_i = \partial^2 F_i/\partial b_i \partial b_i$ be the K by K matrix of second derivatives.

The first Section of the program utilizes the common Steepest Descent procedure, which only uses 1^{st} derivatives. The core subroutine is STEDE with most calculations done in SUBROUTINE SEARCH. Starting with input starting values, the direction of change is the negative of the gradient. New parameter values for the i^{th} change are given by:

$$b_{i+1} = b_i - \lambda_i g_i$$

Each change is called an iteration. The distance of change along the gradient, λ_i, is called the abscissa. The selection of λ_i is determined by an interpolating procedure in SUBROUTINE INTPOL. The selection of λ_i is to choose that λ such that the inner product of g_i and the new gradient g_{i+1} evaluated at the new b_{i+1} is zero. The inner product is called the slope. Program tolerances determine when the slope is "close" to zero. Each attempt by the program to interpolate λ is called a try. The program continues Steepest Descent until a stop condition is encountered. The function and its derivatives are evaluated in SUBROUTINE FCTGR. Errors in FCTGR can cause stoppage. Also, errors "direction of search uphill"; or "No convergence in SEARCH" are both indicators that the program has been unable to meet slope conditions for an iteration. The program may also achieve convergence in STEDE (unlikely). The most common reason for stopping Steepest Descent is that the function value has leveled off. If, for two iterations, the function value changes less than .05 (set as variable PR in SUBROUTINE VINCEN), the program moves on to Fletcher-Powell.

The Fletcher-Powell iterations (SUBROUTINE FLEPOW) are similar to steepest descent except that they utilize 2^{nd} derivatives. Using parameter values from steepest descent, the program calculates the matrix of analytic second derivatives, G_i (the hessian matrix) in SUBROUTINE SECDRV, and inverts it. If the program inverts the hessian, the hessian code is 0. If the matrix cannot be inverted, the number of the row (element of the parameter set) which caused the problem is printed, along with the implied determinant at that point, and the hessian code is 1. The matrix G^{-1} is used as the starting value for the Fletcher-Powell iteration matrix, H_i. If G_i cannot be inverted, an identity matrix is used as a starting value. In Fletcher-Powell, instead of just using g_i, new parameter values are given by:

$$b_{i+1} = b_i - \lambda_i H_i g_i,$$

with b_i, λ and G_i as defined in STEDE. The selection of λ_i is identical to STEDE. Starting with G^{-1}, the matrix H_i is modified for each iteration by:

$$H_{i+1} = H_i + \frac{(b_{i+1} - b_i)(b_{i+1} - b_i)'}{(b_{i+1} - b_i)'(g_{i+1} - g_i)} - \frac{(H_i[g_{i+1} - g_i])(H_i[g_{i+1} - g_i])'}{[g_{i+1} - g_i]'H_i[g_{i+1} - g_i]}$$

The program stops in error for one of 5 reasons:

1. Function error in FCTGR. This can occur for a number of reasons, including: parameters such as correlations or variances in an inadmissible range; failure to converge when calculating a fixed effect; too large (or small) an observation value when computing a cumulative normal for a Tobit or probit; or too large an exponential value when computing a logit. Cutoffs for these decisions are set in the MAIN program (EMAX and FMAX) and SUBROUTINE ERRWRT. It is possible to override the cutoff and continue execution by use of the command OUTLY.

2. Failure to meet slope conditions in SEARCH in the number of "tries" allowed per iteration.

3. Direction of search uphill. This latter condition can occur if $g_i'H_ig_i$ is nonpositive, and is indicative of a poor H_i matrix.

4. The program does not converge in 50 (input as variable ITERA) iterations.

5. The program runs out of time (input as variable SECS).

The program also stops when it achieves convergence. There are three possible criteria of convergence:

1. When <u>every</u> element of g_i is less in absolute value than .00001 (input as variable GTOL).

2. If the absolute proportional change ($[F_i - F_{i-1}]/F_i$) in the function value is less than .0000001 (input as variable FTOL) for <u>two</u> straight iterations.

3. When the absolute proportional change in <u>every</u> parameter is less than .0001 (input as variable PTOL) for <u>two</u> straight iterations.

The default convergence criteria is that all three conditions be met. Alternatively, the convergence condition can be that any of the three criteria (or a particular one) be met. Each criteria has its merits and any may be more stringent in some cases. Users may experiment on criteria and tolerances for their own problems. Several words of warning. The gradient criteria (which we like best) is sensitive to scale. It is suggested that users scale parameters to roughly the same size. The parameter criteria is likely to be sensitive to small insignificant coefficients, and perhaps should not be used in these cases. If the program stops without convergence, the vector of first derivatives last computed, g_i, is printed. Element order is identical to printed coefficients. Similarly, if the function value is larger than .00001 (zero is a perfectly identified model) and TYPE = 3, the program will print the value of the or-

thogonality conditions when iterations stop.

When the program achieves convergence, it calls SECDRV and recomputes the analytic hessian and attempts to invert it. If successful, a hessian code of 0 is given. If unsuccessful (hessian code of 1), the last H matrix is used as an approximation of the inverse of the hessian.

Several variables can be changed in SUBROUTINE VINCEN. The number of "tries" per iteration is set to 10 as variable MAXTRY. The variables, TRIAL1, FACT, AAA, and BB are used to interpolate the distance along the gradient. The leveling tolerance, PR, can also be changed.

Program printout optionally (see IPRINT) shows much of what we have just discussed. for each iteration, the program print the absolute proportional change in the function, the maximum gradient element, and the largest absolute proportional change for a parameter element. For the gradient and parameter values, the program prints the parameter number which had the largest element. This is a guide to which parameters are causing trouble in convergence. Summary statistics of the convergence process will also be printed, giving the reason for stoppage.

Several words of warning in using this (or any) numerical optimization routine. First of all "scaling" tends to be very important. In particular, large discrepancies in the magnitudes of parameters and/or in the magnitudes of the orthogonality conditions imposed may hinder finding a legitimate optimum. Frequently these numerical problems can be overcome by rescaling schemes. A provision in the program (see SCALE) is available such that starting values for each parameter can be rescaled to 1. This is particularly important for tolerances on the gradients for convergence. Especially in the estimation of single nonlinear equations and/or systems of equations which included nonlinear restrictions on parameters, one may also want to rescale the equations and/or instruments used so that the initial estimates of the orthogonality conditions are not radically different in magnitudes. Secondly, one should be wary of convergence when the hessian will not invert. This may be a sign that convergence has not really be achieved. Be particularly wary of convergence in a very small number of iterations. Try different starting values or convergence criteria and see if it converges to the same point.