ANNALES DE L'INSEE – N° 30-31/1978

A general framework for the time series-cross section estimation

Bronwyn H. HALL *

This paper presents the design of a program to handle the specific estimation problems associated with time series-cross section data. In order to minimize the costs of dealing with this kind of data, the program design relies on the forming of the appropriate moment matrices by summing over one or the other dimension, weighting or not, as desired. It is shown how this method may be used to estimate several different models, such as OLS with individual constant terms, GLS, Balestra-Nerlove, and the variable coefficients model of Mundlak. In addition, the method is extended to a nonlinear model, the binary probit model with variance components.

* Harvard University. Financial support was provided by the National Science Foundation and the Hoover Institution. I am grateful to Gary Chamberlain, Zvi Griliches, Robert E. Hall, Jerry Hausman, and Yair Mundlak for helpful discussions.

I. INTRODUCTION

Estimation with time series-cross section data is now generally done by researchers using conventional regression packages. The constraints of such programs usually mean that the user is restricted to methods which pool the data, treating all observations alike. If he wishes to use dummy variables to model the differences among individuals, firms, or regions, the number of variables needed usually exceeds the space available in his program, and he must "sweep" them out before running the regression. In any case, he finds himself very restricted in the kinds of models he can estimate which exploit the two-dimensional nature of his data. The design of a program specifically for time series-cross section data, TSCS, is presented here. This program is capable of estimating models with fixed effects or random effects; such models may be the usual linear ones or they may be nonlinear. The specific application of a nonlinear model which is presented is that of the binary probit model, where the dependent variable is a 0-1 dummy variable. In addition, it is shown how the program can be extended to handle the problem of variable coefficients described by Mundlak (5), where the slopes as well as the intercepts are allowed to vary over individuals.

The important simplifying assumption of TSCS is that while intercepts and slopes are allowed to vary across the cross section dimension, they remain constant over time. This means that the data may be summarized as a set of N moment matrices, one for each individual,[1] and the N group mean vectors. Obviously for many panel data problems (data on individuals over several years) this amount of data will not fit into the computer memory usually available. TSCS is designed with this in mind and does not form all these moment matrices at once; it assumes that the data have been organized on tape with the time dimension varying most rapidly (the observations of different years for an individual are adjacent) and it passes through the tape forming sums of the N moment matrices and sums of the moments of the group means. As an option, these may be weighted sums, which enables the program to handle the problem of heteroskedasticity of the residuals, as well as data which are unbalanced (a different number of years observed for each person). If the number of time units is small, the weight for the moments may be a T by T

matrix; this would allow for GLS estimation to correct for serial correlation, for example.

II. ESTIMATION OF THE LINEAR MODEL

The basic linear model is specified as follows:

$$(1) \qquad y_{it} = X_{it}\beta + \alpha_i + u_{it} \qquad i=1,\ldots,N \quad t=1,\ldots,T$$

$$u_i \sim N(0, \sigma_i^2 I_T).$$

The variance of u_i, σ_i^2, may or may not be constant across individuals. The α_i are the individual intercepts which can have a distribution imposed on them, yielding the variance components model, or can be left free to be estimated as individual constants. As we shall see below, some of the assumptions of this model can be relaxed and it will still be estimable in the computer program described here. These assumptions include the linearity of the model, the homoskedasticity of the residuals for the ith individual, and the constancy of β across the individuals. In this section I will describe the estimation of the simple model shown above, and later sections will show how to generalize the program for more complicated models.

In the following I assume that there exist data on N time series, each T in length.[2] The data are stacked in the following manner:

$$y' = [y_1' \quad y_2' \quad . \quad . \quad . \quad y_N']$$

where each y_i is a length T column vector, and similarly for the X_i. There are K independent variables X, including the constant. A TSCS subroutine is available to form the overall moment matrix X'X (including y'y and X'y) as well as the moments of the group means, $\sum_i T_i \bar{X}_i' \bar{X}_i$. Some desirable estimators which can be obtained with these matrices are described below:

1. Constant coefficients with no individual differences among groups (pooled OLS)

Obviously, this estimator is available in any standard regression package. It is included here mainly because it might be wanted for comparison to the more complex estimators. The model is

(2) $y_{it} = X_{it}\beta + u_{it}$ $\quad i=1,\ldots,N;\ t=1,\ldots,T$

where $u \sim (0, \sigma^2 I_{NT})$.

The appropriate estimator is

(3) $b_0 = (X'X)^{-1}X'y$

$$\sigma^2 = \frac{1}{(NT-K)}(y'y - y'Xb).$$

The only moments needed are those over the entire sample for X'X, X'y, and y'y.

We can relax the assumption of constant variance of the residuals across the groups:

$$u \sim (0, \Sigma)$$

$$\Sigma = \text{diag}(\sigma_i^2 I_T).$$

Then the appropriate estimator is the GLS estimator

(4) $b_g = (X'\Sigma^{-1}X)^{-1} X'\Sigma^{-1}y$

which reduces to weighted least squares in this case because of the diagonality of Σ:

(5) $$b_g = \left[\sum_{i=1}^{N} \sigma_i^{-2} (X'X)_i\right]^{-1} \sum_{i=1}^{N} \sigma_i^{-2}(X'y)_i.$$

This estimator requires estimates of the variances σ_i^2 and also the N individual moment matrices $(X'X)_i$. The variance estimates σ_i^2 can be obtained by using the pooled OLS estimates above and the GLS estimator calculated as a second stage, which requires another pass through the data. Under the assumption of the model, this estimator is BAN. Alternatively, the user can supply estimates of the N variances or variance matrices as weights to be used in the regression.

2. Fixed Effects Model

This model assumes that the slopes are the same for all groups, but that the intercepts differ. The model is

(6) $y_{it} = X_{it}\beta + \alpha_i + u_{it}$, where u_i is again IID.

The α_i's (individual constant terms) can be considered to be coefficients of dummy variables which are 1 for the ith group and zero otherwise. If the model is formulated in this manner and an NT by K+N data matrix X is constructed, the OLS estimator of β and α is obtained:

(7) $$\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \frac{1}{T}X'X & \bar{X} \\ \bar{X} & I_N \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{T}X'y \\ \bar{y} \end{bmatrix}$$

where $\bar{X}$ is the matrix of N group means for K variables and $\bar{y}$ is the vector of N means of the dependent variables. The expressions for b and a simplify to the following:

$$(8) \qquad b_w = (X'X - T\bar{X}'\bar{X})^{-1} (X'y - T\bar{X}'\bar{y})$$

$$a = \bar{y} - \bar{X}b_w.$$

This is the well-known result that the slopes for the fixed effects model can be obtained from the "within-group" regression:

$$b_w = [(X-\bar{X} \otimes e_T)(X-\bar{X} \otimes e_T)]^{-1}[(X-\bar{X} \otimes e_T)(y-\bar{y} \otimes e_T)]$$

where e_T is a column vector of T ones. The variance of this estimator is given by

$$(9) \qquad Var(b_w) = \hat{\sigma}_w^2(X'X - T\bar{X}'\bar{X})^{-1}$$

where

$$\hat{\sigma}_w^2 = \frac{1}{(NT-N-K)} [(y'y-T\bar{y}'\bar{y}) - b_w'(X'y - T\bar{X}'\bar{y})].$$

3. Variance - components estimation

The fixed effects model described above can also be viewed as an error components model where the α_i are to be interpreted not as individual constants, but as random disturbances associated with the ith individual or group. This is the model described by, among others, Wallace and Hussain (6), Balestra and Nerlove (1), and Mundlak (4). The formulas I am using for estimation here are drawn from section 5 of the Wallace and Hussain paper; I have simplified the model to include only a random individual effect, but not a time effect. This is more typical of panel data models, where there are usually only a few years observed for any one individual.

Following Wallace and Hussain, the model in equation (6) may be rewritten as

$$(10) \qquad y_{it} = X_{it}\beta + \varepsilon_{it}, \qquad \varepsilon_{it} = \alpha_i + \omega_{it}$$

where

$$\varepsilon \sim (0,\Sigma), \qquad \Sigma = \sigma_\omega^2 I_{nt} + \sigma_\alpha^2(I_n \otimes J_t)$$

and J_t is a T by T matrix of ones. The appropriate estimator of this model is generalized least squares, which takes account of the heteroskedasticity of ε_{it}. Usually the variances σ_α^2 and σ_ω^2 are not known, however, and must be estimated from the data before being used to form an estimate of Σ. This estimate is then used in the usual way to obtain GLS estimates of the coefficients of the model in equation (9):

$$(11) \qquad b_{vc} = [X' \hat{\Sigma}^{-1} X]^{-1} X' \hat{\Sigma}^{-1} y.$$

This equation may be simplified by using the block structure of $\sum$ when obtaining the inverse:

$$\hat{\Sigma}^{-1} = \frac{1}{\hat{\sigma}_\omega^2}\left[I_{nt} - \frac{\hat{\sigma}_\alpha^2 2}{\hat{\sigma}_\omega^2 + T\hat{\sigma}_\alpha^2}(I_n \otimes J_t)\right].$$

If we let

$$\gamma = \hat{\alpha}_\alpha^2/(\hat{\sigma}_\alpha^2 + \hat{\sigma}_\omega^2/T) \tag{12}$$

then the variance of β may be written as

$$b_{vc} = [X'X - T\,\bar{X}'\bar{X}]^{-1}\,[X'y - T\,\bar{X}'\bar{y}] \tag{13}$$

$$\mathrm{Var}(b_{vc}) = \hat{\sigma}_\omega^2[X'X - T\,\bar{X}'\bar{X}]^{-1}.$$

These formulas are very convenient computationally, since they involve only the overall moments of the data $X'X$ and $X'y$ and the moments of the N group means $\bar{X}'\bar{X}$ and $\bar{X}'\bar{y}$.[3]

There are two alternative ways to estimate the variance components σ_α^2 and σ_ω^2: the limited information and full information methods described by Mazodier (3). The limited information method uses the within regression to obtain $\hat{\sigma}_\omega^2$ (equation (9)) and the between regression to estimate $\hat{\sigma}_\alpha^2$:

$$\hat{\sigma}_\alpha^2 = \frac{1}{(N-K-1)}\left[\bar{y}'\bar{y} - \bar{y}'\bar{X}\,(\bar{X}'\bar{X})^{-1}\,\bar{X}'\bar{y}\right] - \sigma_\omega^2/T. \tag{14}$$

This method ignores the fact that the slope coefficients in both regressions should be the same according to the model. The full information method imposes this restriction by using the residuals from pooled OLS to estimate the variances:

$$\hat{\sigma}_\omega^2 = \frac{1}{(NT-N-K)}\left[y'y - y'X\,b_0 - T(\bar{y}'\bar{y} - 2\bar{y}'\bar{X}\,b_0 + b_0\,\bar{X}'\bar{X}\,b_0)\right] \tag{15}$$

$$\hat{\sigma}_\alpha^2 = \frac{1}{(N-K-1)}\left[\bar{y}'\bar{y} - 2\bar{y}'\bar{X}\,b_0 + b_0\,\bar{X}'\bar{X}\,b_0\right] - \sigma_\omega^2/T. \tag{16}$$

As Mazodier has shown, either of these two estimators for σ_α^2 and σ_ω^2 imply the same properties for the slope estimates b_{vc} when used to compute γ; in particular, b_{vc} is unbiased and asymptotically equivalent to the "true" GLS estimator. They both also have potentially the same difficulty in actual computation: the estimate of σ_α^2 may turn out to be negative since it is obtained as the difference of two variances. This is particularly true when T is small. The only feasible solution to this problem would seem to be to set $\hat{\sigma}_\alpha^2$ to zero, which of course just gives us back pooled ordinary least squares estimates and raises some doubts about the original specification of the model. TSCS flags this problem and goes ahead to compute the OLS estimates.

In estimating the variance components model, TSCS proceeds in general by obtaining both sets of variance estimates, limited information and full information, and presents them for comparison by the user. This creates an obvious programming problem: for pooled OLS and for variance components there must be an overall constant in the model, while for fixed effects (the within regression) such a constant is not identified in the presence of the complete set of N dummies. Other variables which do not change over time for an individual present the same kind of problem in the fixed effects model, but their slopes are identified in random effects. TSCS handles this by including all these variables in the within regression and using a generalized inversion subroutine when computing the regression. This subroutine zeroes the corresponding row and column of the variance matrix when it finds a zero element along the diagonal of the moment matrix when factoring it for inversion. The corresponding estimated coefficient and its standard error are then zero and the program adjusts the degrees of freedom for the regression accordingly by examination of the standard errors. This avoids having to respecify the model and reform the moment matrices for the fixed effects estimation.

We have presented the simple fixed effects and random effects models in some detail in order to demonstrate with a familiar problem how the design of TSCS works with time series-cross section data. In the following section we describe the exact design of the program and then give another example of its use, this time on a less familiar estimation problem, which has probably not been programmed previously for time series-cross section data.

III. DESIGN OF THE PROGRAM

A schematic diagram of TSCS and its subroutines is shown in Figure 1. In the following we will briefly describe the normal program flow of TSCS. The first thing the program does is read in a description of the data file for the run: the variable names and their locations; thereafter the variables can be referred to by name rather than by location. The data file may include many more observations or variables than will actually be used; the only requirement on its structure is that

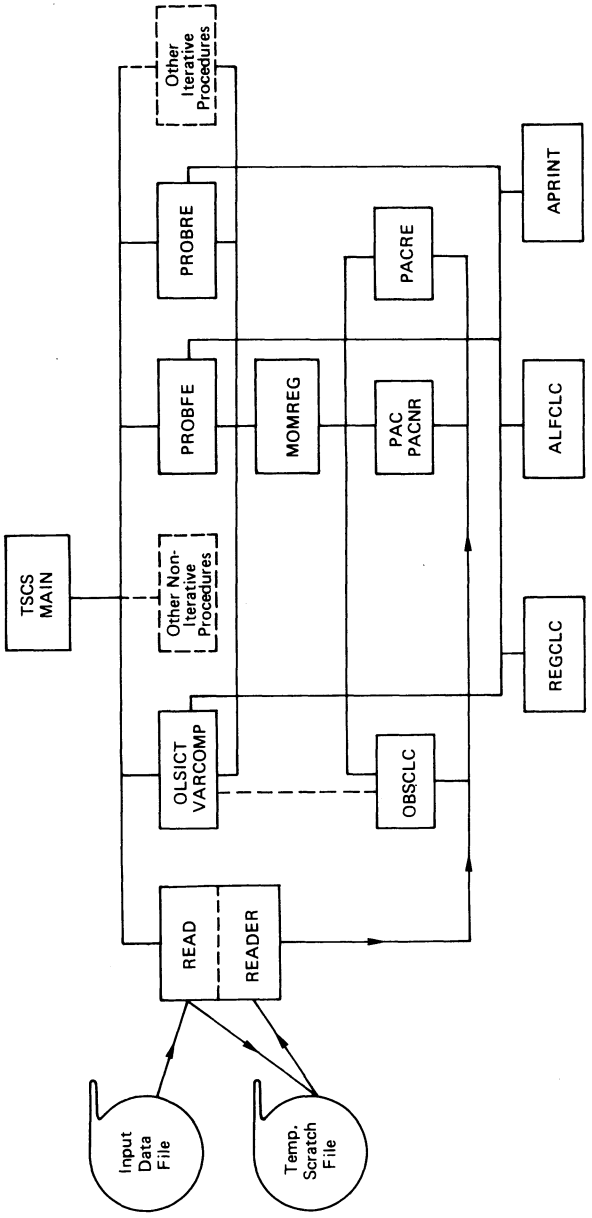

Figure 1: Diagram of the TSCS Program.

the observations for each individual or group be in sequence on the file, with no intervening observations from another group. There is one data file per run, but many different models may be estimated using the same file.

The program then proceeds by reading the name of the procedure desired, any options associated with it, and the list of variables to be used in the current model. Subroutine READ is then called to select these variables off the tape and write a temporary scratch file containing only the observations and variables needed. READ invokes an optional user-written subroutine which provides the user the ability to recode variables or reject observations. This can be extremely convenient when dealing with a large cross section file; however, it is a somewhat inefficient procedure for doing non-iterative regressions, since the scratch file would have to be rewritten every time the specification is changed. There is room for some flexibility of design here; for example, we might wish to define a scratch file containing observations on a larger set of variables for which we wish to form moment matrices and then execute a series of regressions using various submatrices of these moment matrices. This idea is made more concrete when we describe a proposed command language for TSCS in the following section of the paper.

In addition to writing the scratch file of data the READ subroutine is responsible for generating the vector T_i, the number of observations per group, if the user has specified that his data are unbalanced. READ does this by looking for a variable named SERIAL, where all observations with equal SERIAL are assumed to be in the same group. If this variable is not present, READ looks for the variable TIME which is assumed to increase within a group but decrease as soon as a new group begins. If neither variable is present, an error is flagged.

After the scratch file containing the data has been written, TSCS invokes the desired procedure. At present there are four such procedures:

OLSICT	Ordinary least squares with individual constant terms ("within" regression)
VARCOMP	Variance components estimation (this yields also the pooled OLS regression and the "within" and "between" regressions)
PROBFE	Probit estimation with fixed effects
PROBRE	Probit estimation with random effects.

Each of these procedures in turn invokes MOMREG to form the total moment matrix and the moments of the group means; the

arguments to this subroutine are described in detail in Appendix A. MOMREG is capable of passing through a time series-cross section data set and forming both the overall moments and the moments of the group means for a set of specified variables. The moments are obtained in a form suitable for use in a regression subroutine, i.e., the matrix X'X, the vector X'y, and the scalar y'y. The subroutine assumes that the data are organized in a form which lets it determine group means for each group in turn: all years for each individual appear in sequence on the file. The sums of both the total moments and the group mean moments may be optionally weighted. The weights are either supplied to MOMREG by a calling routine or they are created by the subroutine which supplies the data vector. In addition to returning the total moment matrix T_X and the between matrix B_X, MOMREG returns their difference, which may also be weighted:

$$W_X = w_1 T_X - w_2 B_X. \tag{17}$$

This provides for convenient calculation of the fixed effects or random effects linear regression model, for example.

The data are supplied to MOMREG in the following way: the calling program gives MOMREG the name of a subroutine which will supply the current model and this subroutine is invoked by MOMREG NT = $\sum_i T_i$ times to supply the current vector X and the dependent variable y. The subroutine may also supply a weight to be applied to each observation in the forming of the T_X matrix. After each T_i observations MOMREG adds the cross product of the group means to the B_X matrix; a weight for these may also have been supplied by the data subroutine when the current observation is the T_ith observation. The arguments for this subroutine, whose generic name is OBSCLC, are also given in detail in Appendix A. OBSCLC obtains the i,tth data vector by calling an entry point READER in the subroutine READ which reads the temporary scratch file.

The variety of weighting schemes available to MOMREG provide a convenient and flexible way in which to handle both GLS, different weights for different groups, and the unbalanced data problem, where the $(\bar{X}'\bar{X})_i$ are to be weighted by T_i. A new model may be programmed merely by coding a subroutine of the form of OBSCLC to read the data and compute the i,tth observation. The next section of the paper shows how this is done for a particular problem, specifically the maximum likelihood estimation of a binary probit model with fixed and random effects.

After the procedure being executed has obtained the total moments and moments of group means from MOMREG, several utility

routines are available to do the desired estimation and print the results. These subroutines are described in Appendix A. The most important of these is REGCLC, which does a regression calculation, given a matrix Q of order K by K and a vector g of order K by 1:

$$b = Q^{-1}g. \tag{18}$$

REGCLC is also supplied with the sums of squares of the dependent variable and the names of the matrix rows or columns for labelling of output. If the CONSTANT is present in the list, REGCLC can also present the usual regression statistics, such as the R^2, the standard error of the regression, and the mean and standard deviation of the dependent variable. In any case it calculates the sum of squared residuals as $y^2 - gQ^{-1}g$.

When doing estimation with fixed or random effects, the user may be interested in the estimates of the constant terms as well as the slope estimates. For this reason the subroutines ALFCLC, which calculates the vector of constants α_i, and APRINT, which displays them, are provided by TSCS. The equations used by ALFCLC have the following general form:

$$\hat{\alpha}_i = \gamma_i(\bar{y}_i - \bar{X}_i b). \tag{19}$$

γ_i is the weight to be applied to the ith group mean; for fixed effects it is equal to unity and for random effects it is given by equation (12). The calculation of the intercepts α in general requires passing through the scratch data file and should be done only when the user so specifies. In that sense, it is analogous to the calculation and printing of residuals in a conventional regression program. It may also be desirable to present a frequency distribution of the α_i's after they have been calculated instead of, or in addition, to printing them.

We have presented here only a rough outline of the execution of TSCS; the actual subroutines may be linked in a variety of ways depending on the sequence of estimations desired by the user. In a later section of this paper we suggest some ways to specify the desired operations in a convenient manner.

IV. ESTIMATION OF THE BINARY PROBIT MODEL

As an example of how TSCS may be used to program estimation methods other than linear regression, I will describe the design of a procedure which does maximum likelihood probit analysis of covariance. Chamberlain (2) has proposed several iterative methods for estimating the parameters of the following model:

$$(20) \qquad y_{it} = X_{it}\beta + \alpha_i + u_{it} \qquad i=1,\ldots,N; \quad t=1,\ldots,T$$

$$u_{it} \sim N(0, 1).$$

The dependent variable y is not observed, but a dummy variable related to y is:

$$D_{it} = \begin{cases} 1 & \text{if } y_{it} \geq 0 \\ 0 & \text{if } y_{it} < 0. \end{cases}$$

This is the conventional binary probit model with one change, the substitution of the individual constants α_i for one overall constant. This change has the effect of greatly increasing the number of coefficients to be estimated (N+K rather than K+1). The usual way to estimate this model is to apply the Newton-Raphson method to the problem of maximizing the log likelihood function

$$(21) \qquad \ln L = \sum_{D_{it}=1} \ln F(X_{it}\beta + \alpha_i) + \sum_{D_{it}=0} \ln(1 - F(X_{it}\beta + \alpha_i))$$

where F is the cumulative of the normal distribution, with respect to the parameters β and α.

In this particular case, the Newton-Raphson method has the obvious disadvantage of requiring calculation, storage, and inversion of a Hessian of ln L which is of order N+K by N+K, rather than K+1 by K+1. The methods for maximization of ln L proposed by Chamberlain avoid this difficulty by separating the iteration or the parameters into two steps, one an update of the β's, and the second a group by group update of each α_i. This method of breaking up the iteration is similar to that described in the previous section for estimating the linear fixed effects model and can fit easily into the proposed framework of TSCS.

The β iteration proposed by Chamberlain takes the following form:

$$(22) \qquad \beta^{r+1} = \beta^{r} + W_x^{-1} W_{x\psi}^{r}$$

$$W_x = \frac{1}{NT}(X'X - T\bar{X}'\bar{X})$$

$$W^r_{x\psi} = \frac{1}{NT}\,(x^1\psi^r - Tx^1\psi^r).$$

ψ for the (r+1) iteration is the gradient of the log likelihood with respect to the "predicted" y from the rth iteration, $P^r_{it} = X^{\cdot}_{it}\beta^r + \alpha^r_i$.

$$(23) \qquad \psi^r_{it} = \frac{f(P^r_{it})}{F(P^r_{it})}\,D_{it} - \frac{f(P^r_{it})}{1-F(P^r_{it})}\,(1 - D_{it}).$$

It can easily be seen that the expression for the update to the β vector is just a regression calculation for a "within" regression of $\psi_{it} - \bar{\psi}_i$ on $X_{it} - \bar{X}_i$. Therefore, this method may be implemented in TSCS by coding a new subroutine for MOMREG which returns the X_{it}'s of the model and, instead of the y_{it}'s, the current value of ψ_{it} as a function of the X's and the current α's and β's.

The method is iterated by a procedure which loops over the desired number of iterations until convergence of the β's is achieved. The order of the iteration is as follows:

(1) MOMREG is called to form the X'X, X'ψ, $\bar{X}'\bar{X}$, and $\bar{X}'\bar{\psi}$ moments and their differences by invoking the new subroutine, which we shall call PAC (Probit Analysis of Covariance). Note that for this method, the X'X moments need only be formed once, and the appropriate flag should be set in the call to MOMREG to accomplish this.

(2) The regression calculation routine, REGCLC, is called to compute the update vector for β using equation (22).

(3) The α vector is updated with the new change in β, $d\beta^{r+1}$, using the following equation:

$$(24) \qquad \alpha^{r+1}_i = \alpha^r_i + \bar{\psi}^r_i - \bar{X}_i\,d\beta^{r+1}.$$

This is accomplished by the subroutine ALFCLC, the same subroutine used to calculate the individual constant terms in the linear fixed and random effects models; ψ_i replaces y_i and $d\beta$ replaces β in the calling sequence.

(4) Following the calculation of the α's, the β vector is updated by d and convergence is tested for in the usual manner:

$$\left|d\beta^{r+1}_j/\beta^r_j\right| \le \varepsilon \qquad j=1,\ldots,K$$

where ε is a tolerance prespecified by the user.

This iteration technique can be used for any iterative procedure where the iteration is capable of being partitioned into separate updates for β and α. This would include, for example,

an arbitrary nonlinear model with additive individual constant terms.

An alternative to the previous algorithm for estimating the binary probit model with fixed effects is the following scoring algorithm, also due to Chamberlain. Let e_{it} be the "residual" of the observed dummy and the prediction evaluated at the current β^r and α^r:

$$(25) \qquad e_{it} = \frac{D_{it} - F(P_{it}^r)}{f(P_{it}^r)}.$$

We define $\theta_{it} = V(\psi) = E(\partial\psi/\partial P)$ also evaluated at the current parameter values:

$$(26) \qquad \theta_{it} = \frac{f(P_{it}^r)}{F(P_{it}^r)} \quad \frac{f(P_{it}^r)}{(1-F(P_{it}^r))}.$$

Then the scoring algorithm for this model gives the following iteration on β:

$$(27) \qquad d\beta^r = W_X^{r-1} W_{Xe}^r$$

$$W_X^r = \sum_{it} \theta_{it} X_{it} X_{it}' - \sum_i \theta_i \bar{X}_i \bar{X}_i'$$

$$W_{Xe}^r = \sum_{it} \theta_{it} X_{it} e_{it}' - \sum \theta_i \bar{X}_i \bar{e}_i'.$$

$\bar{X}_i$ and $\bar{e}_i$ are the group means of X_{it} and e_{it}, weighted by θ_{it} and $\theta_i = \sum_t \theta_{it}$.

This algorithm may be installed in TSCS by changing the previous variables slightly and using the weighting features of MOMREG. We form the moments of $\theta_{it}X_{it}$ and $\theta_{it}e_{it}$, weighted by $1/\theta_{it}$ and the moments of $\sum_t \theta_{it}X_{it}$ and $\sum_t \theta_{it}e_{it}$ weighting these by $1/\sum_t \theta_{it}$. A new subroutine of the OBSCLC form, PACNR, is written to calculate these terms. The procedure PROBFE can then iterate in the same way as described previously except that the update for the α vector now becomes the following:

$$(28) \qquad \alpha_i^{r+1} = \alpha_i^r + \bar{e}_i - \bar{X}_i \, d\beta^{r+1}$$

where $\bar{X}_i$ and $\bar{e}_i$ are again weighted means.

These two algorithms for probit with fixed effects have been tested on an actual problem with the number of individuals N=1413, T=5, and the number of variables K=2. Both performed fairly well, the second algorithm converging more rapidly than the first. The number of iterations, central processor time, and cost of both on an IBM 370/168 is shown in the table below:

Algorithm	Iterations to convergence	CPU seconds	Cost
EM	16	60	$18.56
Scoring	5	25	8.13

The fact that the EM algorithm took many small steps in the same direction as it neared convergence suggests that its performance might be improved by the use of an algorithm to determine the optimal step in β or α or both. The scoring algorithm, which is a close approximation of the Newton-Raphson method. seems to work quite well with a stepsize equal to unity. The other advantage of this method is that the inverse of W_χ evaluated at $\hat{\beta}$ and $\hat{\alpha}$ corresponding to the maximum of the likelihood function is the estimate of the variance-covariance matrix of $\hat{\beta}$.

V. ESTIMATION OF A VARIABLE COEFFICIENTS MODEL

In this section of the paper we relax the assumption that the slopes (β) are constant over all individuals and allow them to have a dependence on the levels of a particular individual's X's. This leads to the design of a new subroutine for the cumulation of moments, which is similar to MOMREG, but which we shall call MOMVAR. Following Mundlak (5), we write the data matrix X as a partitioned matrix $[Z_i, X_i^r]$ where the last r components of X correspond to coefficients which are the same for all individuals and the first p (the Z_i matrix) correspond to the variable coefficients. There are a total of K=r+p variables in the model. The model for the ith individual can be written:

$$(29) \qquad y_i = (Z_i \; X_i^r) \begin{pmatrix} \beta^p \\ \beta^r \end{pmatrix} + Z_i\alpha_i + u_i$$

or $$y_i = X_i\beta + Z_i\alpha_i + u_i.$$

The α's are given by an auxiliary regression, which describes their dependence on the individual's mean $\bar{X}_i$; the β^p vector represents the overall mean of the variable coefficients and the α_i's are deviations from this mean for each individual:

$$(30) \qquad \alpha_i = (I_p \otimes \bar{X}_i')\pi + w_i.$$

There are p such equations for α. These equations may be combined with equation (28) above to yield Mundlak's estimating

equation (6.4):

$$(31) \qquad y_i = X_i\beta + Z_i(I_p \otimes \bar{X}_i')\pi + Z_i w_i + u_i$$

$$\text{or} \qquad y_i = \tilde{X}_i\gamma + \varepsilon_i .$$

Note that the vector of group means $\bar{X}_i'$ in Mundlak's paper excludes the constant; it is convenient when programming this method to rewrite X_i as

$$(32) \qquad \tilde{X}_i = X_i B_i \quad \text{where} \quad B_i = \begin{bmatrix} I_p \otimes \bar{X}_i' & 0 \\ 0 & I_r \end{bmatrix}$$

and the first p variables in X_i have variable coefficients while the last r are constant over all individuals. The vector $\bar{X}_i'$ contains all K of the variable means; hence $\tilde{X}_i$ is of order T by pK+r.

The error term in the above equation is given by $\varepsilon_i = Z_i w_i + u_i$ where

$$w_i \sim N(0, \Omega_p)$$

and

$$u_i \sim N(0, \sigma_i^2 I_T).$$

The variance-covariance matrix associated with ε_i is therefore

$$(33) \qquad \sum_i = \sigma_i^2 I_T + Z_i \Omega_p Z_i' .$$

Obviously, this matrix violates the assumption of homoskedasticity across the groups which is needed to make ordinary least squares the efficient estimator, so the desired estimator is GLS. In order to use GLS, however, estimates must be obtained for the variances σ_i^2, and for Ω_p, the variance-covariance matrix of the residuals form the auxiliary regressions. Mundlak suggests that the estimates of σ_i^2 and Ω_p be obtained from the OLS estimates of equation (31) above. For convenience when N is large (too large to hold all the data in computer memory) these estimates may also be obtained in terms of the sums of moment matrices to be formed while passing through the data file with TSCS. The moments needed are rather complex notationally; we will denote the r by p submatrix of the ith cross product matrix as $(X'X_{r \times p})_i$ in the following. All of the matrices and vectors involving group means are of order K, the total number of variables in the model:

$$(34)\qquad \tilde{X}'\tilde{X} = \begin{bmatrix} \sum_{i=1}^{N} (X'X_p)_i \otimes (\bar{X}'\bar{X})_i & \sum_{i=1}^{N} (X'X_{p\times r})_i \otimes \bar{X}_i' \\ \sum_{i=1}^{N} (X'X_{r\times p})_i \otimes \bar{X}_i' & X'X_r \end{bmatrix}$$

$$\tilde{X}'\tilde{y} = \begin{bmatrix} \sum_{i=1}^{N} (X'y_p)_i \otimes \bar{X}_i' \\ X'y_r \end{bmatrix}$$

Given these equations, we may calculate the OLS estimate of the pK+r γ's in TSCS by passing through the data file with a new subroutine, MOMVAR, which forms the ith moment matrices X'X and $\bar{X}'\bar{X}$ by calling OBSCLC for the data vector as in MOMREG and then forms the appropriate $\tilde{X}'\tilde{X}$ and $\tilde{X}'\tilde{y}$ matrices by taking Kronecker products of these matrices and cumulating them. If all the coefficients are variable the order of matrices needed is still only K^2 by K^2, and the program is not required to store anything of length N by T or even N.[5]

VI. COMMAND LANGUAGE FOR TSCS

Having described the operation of TSCS in the previous sections of this paper, it remains to specify a convenient language in which the user can describe the procedures he desires to use. The language will be a free format procedure-oriented one like that of the Time Series Processor (7) and many other estimation packages with which economists are familiar. At the present time, each command will be interpreted as it is read and executed immediately; this precludes a user's conditional control of the flow of his program, a restriction which could be eased in the future. All variables on the input file will be referenced by alphameric names (up to 8 characters long on IBM computers) and there are certain reserved variable names which serve special functions. These are the following:

CONSTANT The intercept for the regressions, created by TSCS.

SERIAL The name of a variable which has a unique value for each of the N groups.

TIME — The name of the variable which is an index within each group of the individual observations.

The use of these variables was described in section III; other uses might be imagined.

TSCS has many possible options which modify its operations; some are specific to a particular procedure (local options) and some ar global in nature. The global options are set by an OPTIONS statement, which may appear any place in the input stream, but takes effect only when it appears. There may be more than one of these statements. The global options presently defined are the following:

PRINT (NOPRINT)	Controls the amount of printout desired.
DEBUG (NODBUG)	Provides additional voluminous printout for the use of a programmer.
IFBAL	Specifies whether the data are balanced or not.
NSMALL	Specifies the number of groups for which complete printout of the data as read in and calculated is desired (normally = 0).
N	The number of groups which are currently to be processed in any procedure.

The local options are described below in the descriptions of individual procedures.

The first procedure which must be executed in order to define the file to TSCS is

```
INPUT (options) variable list ;
```

The variable list is a list of the variable names in the order that they are on the input file or a collection of LISTnames, where a LISTname is a name that has been given to a collection of variable names by a LIST statement:

```
LIST listname=variable1, variable2, variable3,.....;
```

INPUT merely defines the input file for the future use of MAKEFILE, but does not cause the data file to be opened or read.

MAKEFILE causes READ to read the input data file, invoke USER if desired, and write the temporary scratch file for use by the regression routines. The form of the instruction is

```
MAKEFILE (USER,other options) variable list ;
```

The list of variables is typically a subset of the complete list defined in the INPUT statement; however, it may include more variables than will be needed in one estimation in order to save on tape reading later. The MAKEFILE statement provides a compromise between the full flexibility of making a new scratch

file created for each regression at a high cost and the restrictiveness of allowing only one specification per TSCS run.

Once the scratch file has been written by READ, the user may call any of the estimation procedures described earlier in this paper. For example,

```
OLSICT (various print options) depvar indepvar1 indepvar2...;
VARCOMP (various print options) depvar indepvar1 indepvar2...;
```

invoke the fixed effects and random effects estimation procedures. Following the first execution of OLSICT, the program will have available several moment matrices which I shall label TXX, TXY, BXX, BXY, WXX, and WXY for total, between, and within moments respectively. Given these matrices it is possible to calculate other regressions of interest, or to present statistics such as the means and variances of variables (assuming that the CONSTANT was included). TSCS should have a set of commands for this purpose, including a PRINT and PUNCH command which can select submatrices from these matrices. In order to make use of such commands, the user must be aware of what is in the matrices TXX, BXX, and WXX following the execution of the OLSICT or VARCOMP commands, since the within matrix, for example, changes between the two procedures.

The approach used in the iterative estimation procedures, such as PROBFE, will be different for efficiency reasons: here an implicit MAKEFILE command will be executed by the statement

```
PROBFE (METHOD= ,LIMIT= ,EPS= ,other options)
       depdummyvar indepvar1 indepvar2 ..... ;
```

The options specify the method of iteration desired, the iteration limit, and the convergence criterion respectively. Only those variables specifically needed by this estimation will be written to the scratch file, to save time during the iterations.

The above represents a bare-bones description of TSCS operations and others can obviously be added (and will be needed). For example, the ability to specify starting values for the iterative procedures is desirable, as is the ability to PUNCH or otherwise process the parameter estimates. It is hoped that TSCS will provide a framework in which new estimation procedures using time series-cross section data can be easily installed without having to be programmed from scratch.

APPENDIX A: DESCRIPTIONS OF THE BASIC SUBROUTINES

1. MOMREG

The arguments to this subroutine are shown in Table A-1. MOMREG loops over the number of individuals, N, and, inside that loop, over the time periods T; at each (i, t) observation the subroutine OBSCLC[6] is called which returns the vector of current X's and Y in DATA. Optionally, this subroutine may also return weights (WEIGHT and WTBAR) for the current matrices which will be used as specified by the option IWT. The moments XX, XY, and YY are cumulated inside the inner loop, along with the data sums for this individual which are temporarily stored in XBAR and YBAR. At the termination of each T loop, the matrices XXBAR, XYBAR, and YYBAR are updated from the current values of XBAR and YBAR using the current value of WTBAR:

```
XXBAR(J,K) = XXBAR(J,K)+XBAR(J)*XBAR(K)*WTBAR.
```

Observe that WTBAR is used whether or not the weighting option is specified, for the following reason: Even if the within moments are not to be weighted for heteroskedasticity, if the data are unbalanced they still must be weighted by the number of observations for the ith individual, T_i. For simplicity of computation, MOMREG uses the sums of X_{it} over t, rather than the corresponding means, and the appropriate equation is then

$$XXBAR = \sum_{i=1}^{N} \frac{1}{T_i} \left(\sum_{t=1}^{T_i} X_{it} \right) \left(\sum_{t=1}^{T_i} X'_{it} \right).$$

The correct value of WTBAR in the unbalanced case is $1/T_i$.

An important feature of MOMREG is that although the IWT option specifies which weights will be input or saved, the subroutine OBSCLC is always permitted to modify these weights; the X weight may be modified at each i,tth observation, while the $\bar{X}$ weight should only be returned for the $t=T_i$th observations. If we call the X weight W and the $\bar{X}$ weight $\bar{W}$, the following table gives the weights which will be received by OBSCLC for the different values of IWT:

IWT	W	$\bar{W}$
0	1.0	1.0
1	WEIGHT(I)	1.0
2	1.0*	1.0
3	1.0	WEIGHT(I)
4	1.0	1.0*
5	WEIGHT	1.0
6	1.0	WEIGHT

*For these two alternatives, the weights returned by OBSCLC will be stored in WEIGHT(I).

Table A-1: Table of arguments to the subroutine MOMREG

Variable name	Input or output	Description
K	input	Number of right-hand-side (X) variables
N	input	Number of individuals (groups)
T	input	Number of time periods per group
XX	output	The K by K moment matrix of the X's (N.T observations)
XY	output	The K by 1 moments of X with Y
YY	output	The sum of squares of Y
XXBAR	output	The K by K moment matrix of the group means (N observations)
XYBAR	output	The K by 1 moments of $\bar{X}$ with $\bar{Y}$
YYBAR	output	The sum of squares of $\bar{Y}$
XBAR	output	A K by 1 vector of overall group means
YBAR	output	The overall mean of Y
XXT	output	The K by K weighted difference of XX and XXBAR
XYT	output	The K by 1 weighted difference of XY and XYBAR
YYT	output	The weighted difference of YY and YYBAR
DATA	working space	A single precision vector, at least K+1 in length, which will be used for temporary storage of the data when reading it in
WEIGHT	input or output	Either an N long vector of weights for the moments or a dummy variable, depending on the value of IWT below
IWT	input	An option variable: 0 No weights are input or to be saved. Weights will still be used, however, if they are supplied by OBSCLC 1 WEIGHT contains weights for the N groups, to be used on the XX matrix 2 Same as 1, but weights are calculated in OBSCLC and are to be stored in WEIGHT

		3 WEIGHT contains weights for the N groups to be used on the X'X matrix
		4 Same as 3, but the weights are to be stored in WEIGHT
		5 There is a scalar in WEIGHT to be used on all X'X.
		6 There is a scalar in WEIGHT to be used on all $\bar{X}'\bar{X}$
OBSCLC	input	An external subroutine name of the routine which calculates DATA, and, optionally, WEIGHT and WTBAR for the i,tth observation

2. OBSCLC

The arguments for this subroutine are given below:

K	input	Number of independent (RHS) variables
I	input	The index of the current individual
T	input	The current time index for this individual
NOBS	input	The total number of time observations for this individual; when T=NOBS the $\bar{X}$ weight should be returned
X	output	A K+1 length vector which is returned by OBSCLC; the first K elements are the X's and the last element is the Y
WEIGHT	input and output	The value of the weight for the i,tth observation
WTBAR	input and output	The value of the weight for the ith group mean

OBSCLC calls a subroutine READER to fetch the current X and Y from a scratch file (this file has been previously written by TSCS to contain only the variables used by the model), does any desired calculations on these variables, and returns them in the vector X. Obviously, these calculations should be restricted to those involving parameters which change during the course of iteration for the sake of efficiency; any recoding or selection of the raw data is done when the scratch file is originally specified to TSCS. As an example of how this subroutine works, an OBSCLC used for fixed and random effects models written in Fortran IV is presented below:

```
      SUBROUTINE OBSCLC(K,I,T,NOBS,X,WEIGHT,WTBAR)
      IMPLICIT REAL*8(A-H,O-Z)
C
C          THIS VERSION OF OBSCLC IS FOR THE FIXED EFFECTS (OLS WITH
C     INDIVIDUAL CONSTANT TERMS) AND RANDOM EFFECTS (VARIANCE COMPONENTS)
C     MODELS FOR BALANCED AND UNBALANCED DATA.
C               BRONWYN H HALL      JULY 1977
C
      REAL X(1)
      INTEGER T
C
      IF(I.GT.1.OR.T.GT.1) GO TO 1
      M=K+1
      WEIGHT=1.0D0
    1 CONTINUE
C
C          READ IN THE X'S AND Y.
C
      CALL READER(I,M,X,&5)
    5 CONTINUE
C
C          IF THIS IS THE LAST OBSERVATION IN THE GROUP ADJUST THE XXBAR
C     WEIGHTS BY THE NUMBER OF OBSERVATIONS IN THE GROUP.
C     FOR FIXED EFFECTS, WTBAR WILL BE 1.0
C     FOR RANDOM EFFECTS, WTBAR WILL BE GAMMA(I).
C
      IF(T.EQ.NOBS) WTBAR=WTBAR/T
      RETURN
      END
```

3. ALFCLC

This subroutine calculates the intercepts for each group by passing through the data file to obtain $\bar{y}_i$ and $\bar{X}_i$ and evaluating the expression $\alpha_i = \lambda_i(\bar{y}_i - \bar{X}_i b)$ where λ_i is a weight for the ith group. The arguments to the subroutine are the following:

K	input	Number of slope coefficients (length of BETA)
N	input	Number of groups or individuals
T	input	Number of time periods per group
ALPHA	input/output	N long vector of individual constant terms
LAMBDA	input	Weight or N long vector of weights for the calculation
XBAR	working space	K long vector for the storage of group means (sums)
BETA	input	K long vector of slope coefficients
IFWTI	input	Logical variable which tells whether the weight supplied in LAMBDA is a vector or constant over all individuals
OBSCLC	input	The external subroutine name of the routine which calculates X_{it} and y_{it}

4. APRINT

APRINT prints the N group intercepts α and the group means of the dependent variable $\bar{y}$. It also calculates the mean and standard deviation of the α vector and prints them. The arguments to the subroutine are the following:

N	input	The number of groups to be printed
ALPHA	input	The N long vector of constant terms
YBAR	input	The N long vector of group means
IFPRNT	input	Logical variable which tells whether α and $\bar{y}$ are to be printed; the mean and standard deviation of α are always printed

5. REGCLC

This subroutine does the regression of y on X, using only the moments of X and y. It can be used to do any regression-type calculation of the form $Q^{-1}g$, although not all of the statistics it calculates may be meaningful in that case. The inversion of Q, which is assumed to be a symmetric matrix stored in its lower triangle, is done by a subroutine[7] which computes the generalized inverse when Q is singular (or near singular); when this has happened, it is signalled by the presence of one or more standard errors equal to zero in the table of regression coefficients. The zero standard errors correspond to the rows and columns of Q which have been dropped in doing the inversion. The arguments to REGCLC are shown below:

K	input	Number of right-hand-side variables (order of XX)
NOBS	input	Total number of observations used in forming moments
XX (Q)	input/output	On input, the K by K moments of the right-hand-side variables K On output, the K by K estimate of the variance-covariance of b
XY (g)	input	K by 1 vector of moments of X with the dependent variable y
YY	input	Sum of squares of the dependent variable
B	output	K long estimate of slope coefficients = $(XX)^{-1}XY$
SSR	output	Sum of squared residuals from the regression = YY - (XY)'B
SIGMA	output	Standard error of the regression = $\sqrt{SSR/(NOBS-K)}$
XNAME	input	K long vector of 8 character labels for the right-hand-side variables
IFPRNT	input	Logical variable which tells whether results are to be printed

FOOTNOTES

[1]Throughout this paper the unit of observation is assumed to be the individual observed over time, although the program could obviously be used for firms, industries, regions, etc. Also, time need not be the other dimension; the program could handle grouped data, for example, data on family members.

[2]It is fairly straightforward to generalize these formulae to problems with unbalanced data ($t=1,\ldots,T_i$ observations per individual), but it complicates the notation, so I have shown the equations for balanced data here, although the program handles unbalanced data as well. The basic difference in the equations is that the cross products of the group means must be weighted by T_i when they are summed; this is a simple extension of the moment matrix weighting subroutine.

[3]When the data are unbalanced, things are not quite so simple; in that case the moments are weighted, rather than unweighted sums of the individual moment matrices. In TSCS, however, this is fairly easy to accomplish using the subroutine MOMREG. The exact equations are the following:

$$b_{VC} = [X'X - \sum_i T_i\gamma_i(\bar{X}'\bar{X})_i]^{-1} \quad [X'y - \sum_i T_i\gamma_i(\bar{X}'\bar{y})_i]$$

where

$$\gamma_i = \hat{\sigma}^2/(\hat{\sigma}_\alpha^2 + \hat{\sigma}_\omega^2/T_i).$$

[4]The estimation methods for this case are drawn from Mundlak (5).

[5]In the case of unbalanced data, it would be convenient to keep in core the vector T_i, which tells how many observations there are in a group of data.

[6]OBSCLC is the generic name of the subroutine; TSCS supplies several different versions of this subroutine as an argument to MOMREG. The versions now available are OBSCLC for the usual linear fixed effects and random effects models, PAC for the Probit Analysis of Covariance by the EM method, PACNR for probit by scoring or the Newton-Raphson method, and PACRE for probit with random effects.

[7]The subroutine is YINVG from the Time Series Processor, and is described in the TSP Version 2.8 User's Manual (7).

REFERENCES

(1) Balestra, Pietro, and Marc Nerlove, "Pooling Cross Section and Time Series Data in the Estimation of a Dynamic Model: The Demand for Natural Gas," Econometrica, 34, 3: 585-612 (July 1966).

(2) Chamberlain, Gary, private communication.

(3) Mazodier, Pascal, "The Econometrics of Error Components Models," Ph.D. thesis, Harvard University, 1971.

(4) Mundlak, Yair, "Estimation of Production and Behavioral Functions from a Combination of Cross-Section and Time Series Data," in Carl F. Christ et al. (eds.), Measurement in Economics, Stanford: Stanford University Press, 1963, pp. 138-166.

(5) Mundlak, Yair, "Models with Variable Coefficients--Integration and Extension," Harvard Institute of Economic Research Discussion Paper No. 499.

(6) Wallace, T. D., and A. Hussain, "The Use of Error Components Models in Combining Cross Section with Time Series Data," Econometrica, 37, 1: 55-72 (January 1969).

(7) Hall, Bronwyn H., TSP User's Manual, Version 2.8, Harvard Institute of Economic Research Technical Paper #12, December 1976.