gretl working papers

DPB: Dynamic Panel Binary data models in Gretl

Riccardo Lucchetti
Università Politecnica delle Marche
r.lucchetti@univpm.it

Claudia Pigini
Università Politecnica delle Marche
c.pigini@univpm.it

working paper #1

April 27, 2015

Abstract

This paper presents the Gretl function package DPB for estimating dynamic binary models with panel data. The package contains routines for the estimation of the random-effects dynamic probit model proposed by Heckman (1981b) and its generalisation by Hyslop (1999) and Keane and Sauer (2009) to accommodate AR(1) disturbances. The fixed-effects estimator by Bartolucci and Nigro (2010) is also implemented. DPB is available on the Gretl function packages archive.

Keywords: Gretl function package, Random-Effects Dynamic Probit model, Quadratic Exponential model, Gauss-Hermite quadrature, simulated Maximum Likelihood, Conditional Maximum Likelihood.

JEL codes: C23, C25, C63

1 Introduction

Non-linear dynamic models for binary dependent variables with longitudinal data are nowadays quite common in microeconometric applications, especially given the increasing availability of panel datasets. One of the most attractive features of models of the type

$$\begin{aligned} y^*_{it} &= \gamma y_{it-1} + \boldsymbol{x}'_{it}\boldsymbol{\beta} + \alpha_i + \varepsilon_{it} \\ y_{it} &= 1\{y^*_{it} \geq 0\} \quad \text{for} \quad i = 1, \ldots, n \;\; t = 2, \ldots, T \end{aligned}$$

is that they lend themselves very naturally to an interpretation in terms of *true state dependence*, that is a situation in which the *realisation* of an event affects the *probability* of the same event occurring in the future, as opposed to simple time persistence in y^*_{it}, which may be due to covariates $\boldsymbol{x}_{it}$ and/or to the time-invariant unobserved heterogeneity (Heckman, 1981a).

This feature is particularly attractive when the problem at hand displays some form of path dependence. Therefore, these models have been found to be extremely useful in the analysis of several microeconomic topics: labour force participation, more specifically female labour supply (Heckman and Borjas, 1980; Hyslop, 1999; Carrasco, 2001; Arulampalam, 2002; Stewart, 2007; Keane and Sauer, 2009), self-assessed health condition (Contoyannis et al., 2004; Heiss, 2011; Halliday, 2008; Carro and Traferri, 2012), poverty transitions (Cappellari and Jenkins, 2004; Biewen, 2009), unionisation of workers (Stewart, 2006), welfare participation (Wunder and Riphahn, 2014), remittance decisions by migrants (Bettin and Lucchetti, 2012), and access to credit (Alessie et al., 2004; Brown et al., 2012; Giarda, 2013; Pigini et al., 2014).

While static models are relatively mainstream and are supported by most statistical and econometric software, dynamic models are more complex to implement and, therefore, estimation routines are not always readily available to the practitioner. Compared to their static counterparts, dealing with unobserved heterogeneity in these models raises several complex issues, both statistical and computational in nature. The main statistical problem lies in the so-called "initial conditions problem": clearly, the recursive nature of the model calls for some kind of conditioning when writing the log-likelihood. This, however, is made problematic by the existence of the time-invariant unobserved individual effects α_i, which is correlated with the initial observation.

Random-Effects (RE henceforth) approaches tackle the initial conditions problem by modelling the joint distribution of the outcomes conditional on y_1. Historically, the first proposal is due to Heckman (1981b) who, building on the static RE estimator, proposed a model for the joint distribution for the response variable $\boldsymbol{y}_i = [y_{i1}, \ldots, y_{iT}]$, in which a separate reduced-form model for the initial observation y_{i1} is approximated via a linearised index function. Under suitable distributional assumptions, the joint log-likelihood for $\boldsymbol{y}_i$ may be evaluated by means of Gauss–Hermite quadrature (Butler and Moffitt, 1982) and estimation may be carried out by Maximum Likelihood (ML).

Generalisations of Heckman's estimator were later proposed by Hyslop (1999), who introduced autoregressive error terms, and Keane and Sauer (2009), who further generalised it to a model with a more flexible treatment of the initial condition equation. In these cases, however, it is necessary to evaluate multivariate normal integrals with an arbitrary correlation structure among error terms and unobserved heterogeneity. This calls for simulation techniques such as GHK (Geweke, 1989; Hajivassiliou and McFadden, 1998; Keane, 1994), which make estimation considerably more costly in terms of CPU requirements.

Alternatively, Wooldridge (2005) proposed modelling the distribution of individual unobserved effect α_i conditional on y_1 and on the history of covariates instead of dealing with the joint distribution of all outcomes.[1] Wooldrigde's estimator employs techniques for dealing with the initial observation problem in such a way that estimation can be carried out through ordinary RE probit routines with the addition of some ad-hoc explanatory variables. This alternative has become extremely popular, but unfortunately does not lend itself to a natural generalisation to the case with autocorrelated disturbances, which is often called for in practice.

The initial conditions problem can also be circumvented via a Fixed-Effects (FE) approach, which makes it possible to estimate the regression parameters consistently without having to make distributional assumptions on the unobserved heterogeneity. The key idea is to condition the joint distribution of $\boldsymbol{y}_i$ on a suitably defined sufficient statistic for α_i. For static logit models, it is possible to define a Conditional Maximum Likelihood (CML) estimator. Its dynamic extension, however, has not gained widespread adoption in empirical work since it cannot be easily generalised to every time-configuration of the panel and requires strong restrictions to the model specification. Moreover, these models generally require that at least a transition between the states 0 and 1 is observed for the individual to contribute to the likelihood. As a result, the number of usable observations often reduces drastically compared to the sample size, especially in the cases when strong persistence in the dependent variable is exactly the reason why a dynamic model is needed.

The first proposal of a FE logit model can be found in Chamberlain (1985): estimation relies on conditional inference and, therefore, is rather simple to perform. Exogenous covariates, however, cannot be included and the proposed sufficient statistic for incidental parameters needs to be determined on a case-wise basis according to the time-series length. Honoré and Kyriazidou (2000) extended Chamberlain's formulation in order to include explanatory variables; this approach, however, implies the usage of semi-parametric techniques that require a substantial computational effort. Moreover, time-dummies (a customary addition to practically all empirical models) cannot be handled either. Recently, Bartolucci and Nigro (2010) defined a dynamic model, which belongs to the quadratic exponential family and resembles closely a dynamic logit model, which overcomes most of the difficulties encountered by Honoré and Kyriazidou's estimator and can be implemented by suitably adapting ordinary static FE logit software.[2]

In this work, we present the `Gretl` implementation of the available set of tools to estimate dynamic models for binary dependent variables in panel datasets by both FE and RE approaches, collected in the `DPB` (*Dynamic Panel Binary*) function package. The RE models contained in `DPB` are the dynamic probit with linearised index initial condition proposed in Heckman (1981b) and the generalisations by Hyslop (1999) and Keane and Sauer (2009). Compared to the other available estimators based on a RE approach, Heckman's estimator has been shown to suffer from remarkably little small-sample bias (Miranda, 2007; Akay, 2012) and is widely used in microeconomic applications. On the contrary, the estimator proposed by Wooldridge (2005) is not included in the package *per se*, despite its common use by practitioners, since `Gretl` already provides suitable functions natively: therefore, we just provide an example showing how to implement it via a simple script. Finally, `DPB` also contains the software for estimating the quadratic exponential model in Bartolucci and Nigro (2010).

[1] Similar approaches have been proposed by Orme (1997, 2001) and Arulampalam and Stewart (2009) which we refer to for a more detailed discussion.

[2] Estimators based on the FE approach for long panels ($T \to \infty$) have also been proposed, among which Hahn and Newey (2004), Carro (2007), Fernández-Val (2009) Hahn and Kuersteiner (2011), Bartolucci et al. (2014). Here we focus only on short panels ($n \to \infty$), which are the datasets usually available to applied microeconomists.

The rest of the paper is organised as follows: Section 2 first lays the methodological background for the estimators provided in the package, while in Section 3 we discuss the main computational issues of their implementation; Section 4 describes in detail the features of DPB; Section 5 provides an empirical application based on a dataset of unionised workers extracted from the U.S. National Longitudinal Survey of Youth; Section 6 concludes.

2 Methodological background

A general model for the conditional probability of a binary response variable y_{it} can be written as

$$p\left(y_{it}|\mathscr{F}_t, \alpha_i; \boldsymbol{\psi}_0\right) \quad \text{for} \quad i = 1, \ldots, n \;\; t = 1, \ldots, T,$$

where $\mathscr{F}_t$ is the information set at time t available to individual i, which, in general, may consist of a set of individual covariates $\boldsymbol{X}_i = [\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{iT}]$ as well as the lag of the response variable $\boldsymbol{y}_i^{t-1} = [y_{i1}, \ldots, y_{it-1}]$, α_i is the individual time-invariant unobserved effect, which is assumed to be a continuous r. v.; $\boldsymbol{\psi}_0$ is the vector of model parameters. In the simple case where the relevant conditioning information set is $\boldsymbol{X}_i$, i. e. y_{it} does not depend on $\boldsymbol{y}_i^{t-1}$, the joint probability of $\boldsymbol{y}_i = [y_{i1}, \ldots, y_{iT}]$ for individual i is

$$p\left(\boldsymbol{y}_i|\boldsymbol{X}_i, \alpha_i; \boldsymbol{\psi}_0\right) = \int_{\mathbb{R}} \prod_{t=1}^{T} p\left(y_{it}|\boldsymbol{X}_i, \alpha_i; \boldsymbol{\psi}_0\right) \mathrm{d}F(\alpha_i).$$

In a dynamic model y_{it} is allowed to depend on its past history. When unobserved effects are present, the process for the response variable needs to be initialised in order to account for how $\boldsymbol{y}_i$ relates to the process before the observations started being available. The dependence on the past of y_{it} is usually likely to be limited to its first lag, so that the relevant subset of $\mathscr{F}_t$ is $[y_{it-1}, \boldsymbol{X}_i]$. Therefore, the joint probability of $\boldsymbol{y}_i$, conditional on $\boldsymbol{X}_i$ and α_i, becomes

$$p\left(\boldsymbol{y}_i|\boldsymbol{X}_i, \alpha_i; \boldsymbol{\psi}_0\right) = \int_{\mathbb{R}} p(y_{i1}|\boldsymbol{X}_i, \alpha_i; \boldsymbol{\psi}_0) \prod_{t=2}^{T} p\left(y_{it}|y_{it-1}, \boldsymbol{X}_i, \alpha_i; \boldsymbol{\psi}_0\right) \mathrm{d}F(\alpha_i).$$

The rest of the section describes the RE approach proposed by Heckman (1981b) and its generalisations and the FE approach put forward by Bartolucci and Nigro (2010). In the first case, a reduced form equation for $p(y_{i1}|\boldsymbol{X}_i, \alpha_i; \boldsymbol{\psi}_0)$ is specified. In the second case, the unobserved effect α_i is eliminated by conditioning $p\left(\boldsymbol{y}_i|\boldsymbol{X}_i, \alpha_i; \boldsymbol{\psi}_0\right)$ on a suitable sufficient statistic and, as a result, the initial condition does not need to be dealt with, so that the joint probability can be written as $p\left(\boldsymbol{y}_i|\boldsymbol{X}_i, y_{i1}, \alpha_i; \boldsymbol{\psi}_0\right)$.

2.1 Random-Effects approach

The estimator proposed by Heckman (1981b) is based on a standard formulation of a dynamic RE binary choice model with an additional equation for the initial observation y_{i1}:

$$y_{it} = 1\{\gamma y_{it-1} + \boldsymbol{x}_{it}'\boldsymbol{\beta} + \alpha_i + \varepsilon_{it} \geq 0\} \quad \text{for} \quad i = 1, \ldots, n \;\; t = 2, \ldots, T \tag{1}$$

$$y_{i1} = 1\{\boldsymbol{z}_{i1}'\boldsymbol{\pi} + \theta\alpha_i + \varepsilon_{i1} \geq 0\} \quad \text{for} \quad i = 1, \ldots, n \tag{2}$$

where y_{it} is the binary response variable, $1\{\cdot\}$ is an indicator function, $\boldsymbol{x}_{it}$ is a vector of individual covariates and $\boldsymbol{z}_{i1}$ contains the values of $\boldsymbol{X}_i$ in the first period and pre-sample information that can be used as exclusion restrictions. The assumptions on α_i and $\boldsymbol{\varepsilon}_i$ are: $\mathrm{E}\left[\varepsilon_{it}|\boldsymbol{X}_i, \alpha_i\right] = 0$; orthogonality of α_i and $\boldsymbol{X}_i$, $\mathrm{E}\left[\alpha_i|\boldsymbol{X}_i\right] = 0$; joint normality conditional on $\boldsymbol{X}_i$ $\left[\theta\alpha_i + \varepsilon_{i1}, \alpha_i + \varepsilon_{i2}, \ldots, \alpha_i + \varepsilon_{iT}\right]' \sim N\left(\mathbf{0}; \boldsymbol{\Sigma}\right)$ with

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1+\theta^2\sigma_\alpha^2 & \theta\sigma_\alpha^2 & \theta\sigma_\alpha^2 & \ldots \\ \theta\sigma_\alpha^2 & 1+\sigma_\alpha^2 & \sigma_\alpha^2 & \ldots \\ \theta\sigma_\alpha^2 & \sigma_\alpha^2 & 1+\sigma_\alpha^2 & \ldots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{3}$$

where $\sigma_\alpha^2 = V(\alpha_i)$.

In this paper, this model will be referred to as the "DP" model (*Dynamic Probit*). Note that the distributional assumption implies independence between α_i and $\boldsymbol{\varepsilon}_i$ and absence of autocorrelation in $\boldsymbol{\varepsilon}_i$, i. e. $\mathrm{E}\left[\varepsilon_{it}\varepsilon_{is}\right] = 0$ for $t, s = 1, \ldots, T$ and $t \neq s$. Under these assumptions, the parameter vector $\boldsymbol{\psi} = \left[\boldsymbol{\beta}', \gamma, \boldsymbol{\pi}', \theta, \sigma_\alpha\right]$ can be estimated by ML, and the i-th contribution to the likelihood is given by the following expression:

$$\mathscr{L}_i(\boldsymbol{\psi}) = \int_{\mathbb{R}} \Phi\left[(\boldsymbol{z}_{i1}'\boldsymbol{\pi} + \theta\alpha_i)(2y_{i1} - 1)\right] \prod_{t=2}^{T} \Phi\left[(\gamma y_{it-1} + \boldsymbol{x}_{it}'\boldsymbol{\beta} + \alpha_i)(2y_{it} - 1)\right] \mathrm{d}\Phi\left(\frac{\alpha_i}{\sigma_\alpha}\right),$$

where $\Phi(\cdot)$ is the standard normal c.d.f. Since the unobserved effects are normally distributed, the integral over α_i can be evaluated by means of Gauss-Hermite quadrature (Butler and Moffitt, 1982).

Hyslop (1999) considered an interesting generalisation of Heckman's approach by allowing for autocorrelation in $\boldsymbol{\varepsilon}_i$. In terms of interpretation, this setting makes it possible to further disentangle two different sources of time persistence: in Heckman (1981b), the *true* state dependence captured by γ in (1) is isolated form the persistence induced by time-invariant unobserved effects α_i; with autocorrelated errors, the persistence in the time-varying unobserved effects is also parametrised. The error terms $\boldsymbol{\varepsilon}_{it}$ follow the AR(1) process

$$\varepsilon_{it} = \rho\varepsilon_{it-1} + \eta_{it} \quad \text{for} \quad t = 2, \ldots, T$$

where $|\rho| \leq 1$ and $\eta_{it} \sim N(0, 1-\rho^2)$. Therefore, the variance-covariance matrix of the error components needs to be modified as follows:

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1+\theta^2\sigma_\alpha^2 & \rho+\theta\sigma_\alpha^2 & \rho^2+\theta\sigma_\alpha^2 & \ldots \\ \rho+\theta\sigma_\alpha^2 & 1+\sigma_\alpha^2 & \rho+\sigma_\alpha^2 & \ldots \\ \rho^2+\theta\sigma_\alpha^2 & \rho+\sigma_\alpha^2 & 1+\sigma_\alpha^2 & \ldots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{4}$$

Note that (4) reduces to (3) for $\rho = 0$; hence, we will call this the "ADP" model (*AR1 Dynamic Probit*).

Recently, Keane and Sauer (2009) introduced a more general version of $\boldsymbol{\Sigma}$ in which an additional parameter is defined. The starting point is to modify (2) as

$$y_{i1} = 1\{\boldsymbol{z}_{i1}'\boldsymbol{\pi} + \theta\alpha_i + u_i \geq 0\},$$

with $E(u_i \cdot \varepsilon_{i2}) = \tau$. Note that since both u_i and ε_{it} need to be normalised for identification, τ is effectively a correlation coefficient, so $|\tau| \leq 1$ holds by the Cauchy-Schwartz inequality.

Therefore, (4) becomes:
$[\theta\alpha_i + u_i, \alpha_i + \varepsilon_{i2}, \dots, \alpha_i + \varepsilon_{iT}]' \sim N(\mathbf{0}; \mathbf{\Sigma})$ with

$$\mathbf{\Sigma} = \begin{bmatrix} 1+\theta^2\sigma_\alpha^2 & \tau\rho+\theta\sigma_\alpha^2 & \tau\rho^2+\theta\sigma_\alpha^2 & \dots \\ \tau\rho+\theta\sigma_\alpha^2 & 1+\sigma_\alpha^2 & \rho+\sigma_\alpha^2 & \dots \\ \tau\rho^2+\theta\sigma_\alpha^2 & \rho+\sigma_\alpha^2 & 1+\sigma_\alpha^2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{5}$$

Notice that (5) is equal to (4) for $\tau = 1$ and equal to (3) for $\rho = 0$. The acronym we will use henceforth for this model is "GADP" (*Generalised ADP*).

In theory, its greater generality makes model (5) more appealing for applied work than it restricted counterpart (4) and, *a fortiori*, (3). However, there are two factors that may make it advisable to opt for the more restrictive versions of the model. The first one is computational complexity: the specification of ε_{it} as an AR(1) process makes it impossible to evaluate the joint probability $p(\cdot)$ by integrating out the random effect α_i. Instead, T-variate normal probabilities must be evaluated by simulation by using the GHK algorithm (Geweke, 1989; Hajivassiliou and McFadden, 1998; Keane, 1994) in order to compute the likelihood function:

$$\mathscr{L}_i^*(\boldsymbol{\psi}) = \frac{1}{R}\sum_{r=1}^{R} \Phi_{Tr}^*(a_i, \boldsymbol{b}_i, \boldsymbol{C})$$

with $a_i = (\boldsymbol{z}_{i1}'\boldsymbol{\pi})(2y_{i1} - 1)$ and $\boldsymbol{b}_i = [b_{i2}, \dots, b_{iT}]$, where $b_{it} = (\gamma y_{it-1} + \boldsymbol{x}_{it}'\boldsymbol{\beta})(2y_{it} - 1)$; $\boldsymbol{C}$ is the lower-triangular Cholesky factor of $\mathbf{\Sigma}$, defined in (4) or (5), and r is the number of random draws used in the simulation. The second one is that, in finite samples, the ADP and GADP models (especially the latter) may suffer from serious identification problems for certain regions of the parameter space; in some cases, this could cause serious numerical problems for conducting inference, because of insufficient curvature of the log-likelihood function. These issues will be described in greater detail in section 3.4.

2.2 Fixed-Effects approach

With the exception of linear probability models, FE approaches to the estimation of dynamic binary choice models are usually based on the dynamic logit formulation. In this case, model (1) is subject to strict exogeneity conditional on α_i and it is assumed that the error terms ε_{it}, $t = 1, \dots, T$, are i.i.d. standard logistic random variables. Therefore, the probability of observing y_{it} can be written as

$$p(y_{it}|y_{it-1}, \boldsymbol{x}_{it}, \alpha_i; \boldsymbol{\psi}) = \frac{\exp\left[y_{it}\left(\gamma y_{it-1} + \boldsymbol{x}_{it}'\boldsymbol{\beta} + \alpha_i\right)\right]}{1 + \exp\left(\gamma y_{it-1} + \boldsymbol{x}_{it}'\boldsymbol{\beta} + \alpha_i\right)}.$$

As is well known (Chamberlain, 1985), in the static case ($\gamma = 0$) the *total score* $y_{i+} \equiv \sum_t y_{it}$ is a sufficient statistic for α_i, on which the distribution of $\boldsymbol{y}_i$ can be conditioned, so as to remove the unobserved individual effect α_i; unfortunately, in the dynamic logit model ($\gamma \neq 0$) equivalent sufficient statistics may exist, bu they must be derived on a case-wise basis. Moreover, the inclusion of explanatory variables requires restrictions on their distribution and a non-negligible computational effort (Honoré and Kyriazidou, 2000).

The joint probability of observing the response configuration $\boldsymbol{y}_i$ conditional on the initial observation y_{i1} is

$$p(\boldsymbol{y}_i|\boldsymbol{X}_i, y_{i1}, \alpha_i; \boldsymbol{\psi}) = \frac{\exp\left(\sum_t y_{it}y_{it-1}\gamma + \sum_t y_{it}\boldsymbol{x}_{it}'\boldsymbol{\beta} + y_{i+}\alpha_i\right)}{\prod_t\left[1 + \exp\left(\gamma y_{it-1} + \boldsymbol{x}_{it}'\boldsymbol{\beta} + \alpha_i\right)\right]} \tag{6}$$

where sums and products go from $t = 2$ to T.

The quadratic exponential model proposed by Bartolucci and Nigro (2010) (for which we will use, from here onward, the "QE" acronym) directly defines the joint probability of $\boldsymbol{y}_i$ as

$$p(\boldsymbol{y}_i|\boldsymbol{X}_i, y_{i1}, \alpha_i; \boldsymbol{\psi}) = \frac{\exp\left(\sum_t y_{it}y_{it-1}\gamma + \sum_t y_{it}\boldsymbol{x}_{it}'\boldsymbol{\beta}_1 + y_{iT}\left(\mu + \boldsymbol{x}_{iT}'\boldsymbol{\beta}_2\right) + y_{i+}\alpha_i\right)}{\sum_{\boldsymbol{b}\in\mathbb{B}} \exp\left(\sum_t b_t b_{t-1}\gamma + \sum_t b_t\boldsymbol{x}_{it}'\boldsymbol{\beta}_1 + b_T\left(\mu + \boldsymbol{x}_{iT}'\boldsymbol{\beta}_2\right) + b_+\alpha_i\right)} \tag{7}$$

where $b_+ \equiv \sum_t b_t$ and $\mathbb{B} \equiv \left\{\boldsymbol{b} : \boldsymbol{b} \in \{0,1\}^T\right\}$, that is the set of all possible T-vectors $\boldsymbol{b}$ containing zeros and ones.

This probability closely resembles the one in (6) and, under (6) and (7), Bartolucci and Nigro (2010) show that γ has the same interpretation in terms of log-odds ratio between each pair of consecutive y_{it}s. It can be shown that the total score y_{i+} is a sufficient statistic for the unobserved effects α_i in (7). The conditional distribution based on the total score is

$$p(\boldsymbol{y}_i|\boldsymbol{X}_i, y_{i1}, y_{i+}; \boldsymbol{\psi}) = \frac{p(\boldsymbol{y}_i|\boldsymbol{X}_i, y_{i1}, \alpha_i; \boldsymbol{\psi})}{p(y_{i+}|\boldsymbol{X}_i, y_{i1}, \alpha_i; \boldsymbol{\psi})} =$$

$$\frac{\exp\left[\sum_t y_{it-1}y_{it}\gamma + \sum_t y_{it}\boldsymbol{x}_{it}'\boldsymbol{\beta}_1 + y_{iT}\left(\mu + \boldsymbol{x}_{iT}'\boldsymbol{\beta}_2\right)\right]}{\sum_{\boldsymbol{b}:\, b_+=y_{i+}} \exp\left[\sum_t b_t b_{t-1}\gamma + \sum_t b_t\boldsymbol{x}_{it}'\boldsymbol{\beta}_1 + b_T\left(\mu + \boldsymbol{x}_{iT}'\boldsymbol{\beta}_2\right)\right]}. \tag{8}$$

So that the denominator contains only those vectors $\boldsymbol{b} \in \mathbb{B}$ such that $b_+ = y_{i+}$. By using the above conditional probability, the conditional log-likelihood can be written as

$$\ell(\boldsymbol{\psi}) = \sum_{i=1}^{n} 1\{0 < y_{i+} < T\} \log p(\boldsymbol{y}_i|\boldsymbol{X}_i, y_{i1}, y_{i+}; \boldsymbol{\psi})$$

and maximised with respect to $\boldsymbol{\psi} = \left[\gamma, \boldsymbol{\beta}_1', \mu, \boldsymbol{\beta}_2'\right]'$. We refer the reader to Bartolucci and Nigro (2010) for details on the score and Hessian of $\ell(\boldsymbol{\psi})$.

3 Computation and numerical issues

3.1 Treatment of missing values

The DPB package can handle unbalanced panels, as long as there are enough consecutive observations for each longitudinal unit. If we use T_i to indicate the maximum time span of consecutive observations for individual i, the requirements for a cross-sectional unit i to be included in the sample are

- $T_i \geq 2$ for the DP, ADP and GADP models;
- $T_i > y_{i,+} > 0$ for the QE models.

For each unit, the longest available consecutive set of observations per individual is used. If more than one sequence is available, we take the most recent one. Of course, the choices above imply the assumption that observability of $y_{i,t}$ and/or $\boldsymbol{x}_{i,t}$ is totally independent of the random variables included in the models. For further details, see Albarrán et al. (2015).

3.2 Computation of the log-likelihood and its derivatives

The two most important algorithms employed for computing the log-likelihood of the four models handled by the package are Gauss-Hermite quadrature for the DP model and the GHK algorithm for the ADP and GADP models.[3] As for the QE model, the main problem lies in efficient computation of the denominator in equation (8), which does not include problematic functions, but is made tricky by the multiplicity of cases to consider. Fortunately, a recursive approach is possible, which we will describe in subsection 3.3.

For Gauss-Hermite quadrature, the `DPB` function package uses the Gretl native function `quadtable`, which guarantees good speed and accuracy. The number of quadrature point can be chosen by the user, with a default of 24. The `DPB` package does not include, at present, the option of using adaptive quadrature methods as recommended, for example, in Rabe-Hesketh et al. (2002).

As for simulation-based methods, `DPB` relies on the Gretl function `ghk`, which natively provides analytic derivatives. This function does not implement optimisation techniques such as the pivoting method by Genz (1992), because Genz's method may introduce discontinuities that would make numerical differentiation problematic. On the other hand, it automatically switches to a parallel implementation on a multi-core machine in a shared-memory environment, which gives a noticeable performance boost on modern CPUs.[4] The default method for feeding the uniform sequence to the GHK algorithm is by using Halton sequences, but the user can switch to the uniform generator used by default in Gretl (the SIMD-oriented implementation of the Mersenne Twister algorithm described in Saito and Matsumoto (2008)) if so desired.

The choice method for optimisation is BFGS with analytical derivatives for the probit models DP, ADP and GADP; this has generally proven quite effective and remarkably more robust and efficient than Newton-Raphson. Alternatively, its limited-memory variant described in Byrd et al. (1995) can also be used. Two of the parameters in the covariance matrix $\mathbf{\Sigma}$ are in fact maximised via an invertible transformation to help numerical stability: σ_α^2 is expressed as its natural logarithm and ρ through the (inverse) hyperbolic tangent transform, so that the parameters that enter the actual maximisation routine are unbounded. Initial values for $\boldsymbol{\beta}$, γ, and $\boldsymbol{\pi}$ are obtained by straightforward linear probability models, while σ_α and θ are both set to 1.

For the QE model, instead, the method of choice is Newton-Raphson, which takes advantage of the fact that the computation of the Hessian matrix is remarkably inexpensive once the analytical score has been been obtained. Initial values for γ, $\boldsymbol{\beta}_1$, μ and $\boldsymbol{\beta}_2$ are simply 0.[5]

3.3 Computation of the denominator term in the QE model

The peculiar nature of the QE model makes it impossible to compute its denominator in a manner akin to the algorithm used in several packages for the ordinary conditional logit model, which is in turn a variation of the recursive algorithm by Krailo and Pike (1984).

The algorithm we implement has the virtue of relegating the recursive computation of the relevant combinations ($\boldsymbol{b}$ vectors) at the initialisation stage; by memorising the rele-

[3]It is perfectly possible to use the GHK technique for estimating the DP model, but it would be rather inefficient. However, it may be interesting to do so for comparison purposes, so the package makes it possible via an option.

[4]Gretl also provides MPI extensions for distributed-memory architectures, but they have not been used in the `DPB` package.

[5]This choice, to our surprise, was found to outperform logit-based initialisation.

vant information into an array of matrices, we avoid recursion at each computation of the likelihood. Our algorithm can be briefly described as follows.

The denominator in equation (8) can be easily written as a function of a matrix Q, whose size and elements are function of three scalars: T_i, y_{i+} and y_{i*}, where $y_{i*} = \sum y_{it} y_{it-1}$ is the number of consecutive ones in $\boldsymbol{y}_i$.

Define an injective index function $j = j(T, y_+, y_*)$ (where we omit the i subscript for conciseness). Obviously, all individuals in the sample with the same values for T, y_+ and y_* can share the same j index. The idea is to pre-generate all the needed Q_j matrices and store them in an array; since $T_i \geq y_{i+} > y_{i*}$ holds, the array can have at most $T^2_{\max}(T_{\max} - 1)$ elements, although in practice the number of distinct elements in the array will be much smaller. In a typical panel data setting, it is unlikely that more than a few hundred distinct Q_j matrices will have to be computed and stored in memory.

The internal structure of Q_j is

$$Q_j = \left[Q^1(T, y_+) | q^2(y_*) \right] \tag{9}$$

in which

1. $Q^1(T, y_+)$ is a matrix with $\frac{T!}{y_+!(T-y_+)!}$ rows and T columns whose rows are the elements of $\mathbb{B}$ (see Section 2.2)

2. $q^2(y_*)$ is a column vector holding the number of times in which you get consecutive ones in the corresponding row of Q^1.

For example, consider an individual for whom $\boldsymbol{y}_i = [0, 1, 1, 0]$; in this case we have $T_i = 4$, $y_+ = 2$ and $y_* = 1$. The corresponding Q matrix would be

$$Q = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

which lists all the possibilities. We call the "relevant" row k_i the one which contains the sequence $\boldsymbol{y}_i$ actually observed (the third row of Q in this example, so $k_i = 3$).

The $Q^1(T, y_+)$ matrices can be computed recursively by noting that

$$Q^1(T, y_+) = \left[\begin{array}{c|c} & 0 \\ Q^1(T-1, y_+) & \vdots \\ & 0 \\ \hline & 1 \\ Q^1(T-1, y_+ - 1) & \vdots \\ & 1 \end{array} \right]$$

with the special cases

$$\begin{aligned} Q^1(n,0) &= [0\dots0] \\ Q^1(n,1) &= I_n \\ Q^1(n,n) &= [1\dots1] \end{aligned}$$

Since none of the necessary Q_j depend on parameters to be estimated, they are precomputed in a preliminary loop and stored in an array of matrices. The algorithm employed can be therefore given the following schematic description:

1. initialise an empty array of matrices Q;
2. for each individual i;
   (a) compute the $j(i)$ index as a function of T_i, y_{i+} and y_{i*};
   (b) if Q_j has already been computed, stop and go to the next individual i; else
      i. compute $Q^1(T_i, y_{i+})$ via the recursive method described above;
      ii. store the relevant row k_i for individual i.
      iii. compute Q_j via equation (9);
      iv. store Q_j into the array at position j.

Once this is done, the likelihood for individual i in (8) becomes a simple function of k_i and Q_j, which don't need to be recomputed during the Newton-Raphson iterations.

3.4 Identification of ρ and τ in short panels for RE probit models with AR(1) disturbances

When estimating RE probit models with autocorrelated errors, maximising the log-likelihood may be quite challenging from a numerical point of view because of weak identification in some of the parameters. Consider first the ADP model, in which the covariance matrix of the compound disturbance terms for a unit is given in equation (4). It is quite obvious that some combinations of the parameters give rise to a very badly conditioned matrix. This happens, in particular, when $\theta \simeq 1$ and $\rho = 1 - \epsilon$ (for "small" ϵ). In those cases, $\mathbf{\Sigma}$ will be very close to be singular, especially in those units for which the time dimension is short (note that all these three possibilities are far from being uncommon in practice). The typical outcome is BFGS taking many iterations to converge or, worse, not converging at all.

The situation is even worse for the GADP model, whose covariance matrix is given in equation (5): first, since the covariance matrix depends on four parameters, an elementary order condition dictates that only the observations for which $T_i \geq 3$ will provide enough curvature to the log-likelihood to separately identify all four parameters, which is not an overly restrictive requirement. However, note that the parameter τ only appears in products like $\tau \cdot \rho^t$, with $1 \geq t \geq (T_i - 1)$. This implies that, for obtaining enough curvature in the objective function in the direction of τ, the sequence $\rho, \rho^2, \rho^3, \dots$ must be noticeably different (from a numerical point of view) from a sequence of constants. In practice, this implies that estimation is likely to fail anytime (*i*) ρ is very close to 0 or to 1 and (*ii*) $T_{\max}$ is small, unless N is truly enormous. In the limit, if $\rho = 0$ or $\rho = 1$ the model is under-identified even when $N \to \infty$ and $T \to \infty$.

4 The DPB function package

In Gretl, function packages are collections of user defined functions made available to other users. The user can download and install the function packages available on the Gretl server by menu or script. A function package can be downloaded and installed simply by invoking the `install` command:

```
Gretl console: type 'help' for a list of commands
? install DPB
Installed DPB
```

Then, in each work session, the function package needs to be loaded by:

```
? include DPB.gfn
```

We refer the reader to the Gretl user guide for illustration on how to download and install function packages from the menu. The function package DPB includes four functions that handle model set-up, option management, estimation and printing of results. A summary of these functions is given in Table 1 and a detailed description is contained in the section "List of Public Functions" of the DPB documentation file.

In the rest of this section, we first illustrate how to estimate the RE models DP, ADP and GADP described in Section 2.1, then we show how to perform estimation of the QE model proposed by Bartolucci and Nigro (2010). All the script files and datasets used in this section are available in the folder *your path*/`.gretl/functions/DPB/examples`, created after the installation of DPB. The minimum version of Gretl required to use the package is 1.10.0.

4.1 Random-Effects dynamic probit

We start by describing how DPB handles the RE approach proposed by Heckman (1981b), by using an artificial dataset generated following (1) and (3):

$$
\begin{aligned}
y_{it} &= 1\{1 + 0.6\, y_{it-1} + 0.5\, x_{it} + \alpha_i + \varepsilon_{it} \geq 0\} \qquad (10)\\
y_{i1} &= 1\{1 + x_{i1} + z_{i1} + 1.2\, \alpha_i + \varepsilon_{i1} \geq 0\},
\end{aligned}
$$

$$
\sigma^2_\alpha = 1, \quad \alpha_i \sim N(0, \sigma^2_\alpha)
$$

$$
x_{it} \sim N(0,1) \quad z_{it} \sim U(0,1),
$$

$$
\text{for} \quad i = 1, \ldots, 4096 \quad \text{and} \quad t = 1, \ldots, 6.
$$

We also assume that $\rho = 0$, so ε_{it} is not autocorrelated. For the purpose of replication, we provide the script file `RE_gendata.inp`: this script generates dynamic binary data under normality with the error structure in (3), sets the parameter values in (10) and stores the artificial data in `DP_artdata.gdtb`. The following code is provided in the `DP_example.inp` script file.

Once the DPB function package has been downloaded form the Gretl server, a simple Hansl script to estimate the DP model is:

```
set echo off
set messages off

include DPB.gfn
open DP_artdata.gdtb
```

Table 1: Summary of functions in DPB

Function	`DPB_setup(string mod, series depvar, list X, list Z[null])`
Return type	`bundle`
Function arguments	`string mod`
	`"DP"`: Dynamic Probit model (Heckman, 1981b)
	`"ADP"`: AR(1) Dynamic Probit model (Hyslop, 1999)
	`"GADP"`: Generalised AR(1) Dynamic Probit model (Keane and Sauer, 2009)
	`"QE"`: Quadratic Exponential model (Bartolucci and Nigro, 2010)
	`series y`: the binary dependent variable
	`list X`: list of the x_{it} in equations (1) and (8)
	`list Z`: list of the z_{i1} in equation (2) (optional)
Description	initialise the model bundle, sub-sample the data as needed, build data matrices, handle default settings
Function	`DPB_setoption(bundle *b, string opt, scalar value)`
Return type	`scalar`
Function arguments	`bundle *b`: pointer to the model bundle
	`string opt`: a string indicating which option to set
	`scalar value`: a scalar value for the option
Description	sets various options of the model (see Table 2); returns an error code, 0 if no error
Function	`DPB_estimate(bundle *bun, matrix *par[null])`
Return type	`void`
Function arguments	`bundle *b`: pointer to the model bundle
	`matrix *par[null]`: matrix holding initial values (optional)
Description	estimates the model
Function	`DPB_printout(bundle *bun)`
Return type	`void`
Function arguments	`bundle *b`: pointer to the model bundle
Description	prints the estimation results

```
setobs id time --panel-vars

list X = const x
list Z = const x z

b = DPB_setup("DP",y,X,Z)
DPB_estimate(&b)
```

The first two lines of code just prevent Gretl from echoing unnecessary output, while the `include` command loads the function package. After setting the panel structure, which is required for `DPB` to work, the lists of explanatory variables are created. In the next line, the call to the public function `DPB_setup` initialises the model and returns a bundle: the first argument is a string containing the name of the model to be estimated (`"DP"` in this case). The remaining arguments `y, X, Z` are the dependent variable, the list of explanatory variables for the primary equation and those for initial condition equation.

The function `DPB_estimate` takes as its argument the pointer to the model bundle and fills the bundle with the estimated quantities. The bundle elements can be listed by simply typing `print b`. A detailed description of the bundle elements is given in the `DPB` documentation. Bundle elements can be accessed by the syntax `npar = b["npar"]` or `npar = b.npar`. The returned bundle can also be stored as an XML file by using the Gretl built-in function `bwrite(b, "mod")` and reloaded, if necessary, by the companion function `b = bread("mod")`. The `DPB_estimate` function also assigns initial values for the parameters; the user can supply a vector of initial values as a second argument to `DPB_estimate`. For instance

```
scalar npar = b.npar
inipar = muniform(npar,1)
DPB_estimate(&b,&inipar)
```

To print out the results, the corresponding public function is called

```
DPB_printout(&b)
```

which produces the following output:

```
Dynamic Probit model
Dependent variable: y
Units: 4096 (observations: 24576)
Covariance matrix: Sandwich
Method: Gaussian quadrature with 24 quadrature points

            coefficient   std. error     z     p-value
  -----------------------------------------------------
  y(-1)      0.615952     0.0338469    18.20   5.34e-74 ***


            coefficient   std. error     z      p-value
  ------------------------------------------------------
  const      0.501168     0.0375208    13.36   1.08e-40   ***
  x          0.502661     0.0138080    36.40   3.73e-290 ***


            coefficient   std. error     z      p-value
  ------------------------------------------------------
  const      1.00756      0.0703784    14.32    1.73e-46 ***
  x          0.940772     0.0483211    19.47    2.00e-84 ***
  z          0.812821     0.113210      7.180   6.98e-13 ***
```

```
           coefficient   std. error     z      p-value
  -----------------------------------------------------
  theta       1.11218      0.0764148    14.55   5.47e-48  ***
  sigma       0.998992     0.0334803    29.84   1.25e-195 ***

  Log-likelihood     -10771.971      AIC      21559.942
           BIC      21624.818      HQC      21580.958

  Wald test = 1325.22 (1 df); p-value = 3.73248e-290
```

where coefficients, standard errors and related z-tests are reported for the state dependence parameter γ, the coefficients of the main equation $\boldsymbol{\beta}$, the coefficients of the initial conditions equation $\boldsymbol{\pi}$, and the covariance matrix parameters θ and σ_α. At the end of the output table, a Wald test for the joint significance of the explanatory variables in the main equation is reported.

As shown in the output headings, the model has been estimated using the Gauss-Hermite quadrature with 24 quadrature points, and the variance-covariance matrix of the parameters has been estimated by a sandwich formula. These are the default settings, which can be altered by the user by calling the public function `DPB_setoption` before `DPB_estimate`. For instance,

```
err = DPB_setoption(&b, "nrep", 32)
DPB_estimate(&b)
```

`DPB_setoption` returns a scalar (called `err` in the above code lines) containing 0 if the option is valid and successfully set, an error code otherwise. For a detailed description of the available options, see Table 2.

In the example above, the string `nrep` and the scalar 32 are used to set the number of quadrature points to be used during estimation.

The DP model can also be estimated by computing the multivariate normal probabilities by GHK instead of using numerical integration by Gauss-Hermite quadrature. After the call to `DPB_setup`, the user can switch the method form GHQ (default) to GHK by calling `DPB_setoption` as follows

```
err = DPB_setoption(&b, "nrep", 100)
err = DPB_setoption(&b, "method", 1)
```

When the GHK method is invoked, a Halton sequence is used by default. The user may instead use random draws from a uniform distribution by setting `"draws"` to 1, as described in Table 2. The number of Halton points used by the GHK algorithm has a default value of 128 that can also be modified by the function `DPB_setoption` with the string `nrep` and the number of points, as illustrated earlier in this section.

In some cases, estimation can be computationally quite intensive. In particular, the GHK algorithm is quite sensitive to the time dimension of the panel, as T sets the dimension for the covariance matrix (3). Table 3 reports the performance of the `DPB` function package for different numbers of quadrature points and GHK replications to give the user an idea of the trade-off between algorithm quality and time. The table was obtained by the following command lines:

```
b = DPB_setup("DP",y,X,Z)
err1 = DPB_setoption(&b, "verbose", 2)
err2 = DPB_setoption(&b, "vcv", 1)
```

Table 2: Description of `DPB_setoption`'s arguments

`string opt`	`scalar value`
`"method"`	0 = Gauss-Hermite Quadrature (GHQ), default choice for the DP model, 1 = GHK algorithm. For the ADP and GADP models, `method` is forced to 1. For the QE model a warning message is printed
`"nrep"`	number of quadrature points or GHK draws. Default is 24 for the DP model with GHQ, 128 for the DP model with GHK and for the ADP and GADP models. For the QE model a warning message is printed
`"vcv"`	parameters covariance matrix 0 = Sandwich (default), 1 = Outer Product of the Gradient (OPG), 2 = Hessian
`"verbose"`	degree of output verbosity 0 = no output is printed, 1 = the log-likelihood at each iterations is printed (default), 2 = log-likelihood, parameters and gradient at each iteration are printed
`"draws"`	type sequence for the GHK algorithm 0 = Halton (default), 1 = Uniform with seed `31415927`. For the DP model with GHQ and the QE model a warning message is printed.

```
loop foreach i 16 24 32
    set stopwatch
    err_$i = DPB_setoption(&b, "nrep", $i)
    DPB_estimate(&b)
    DPB_printout(&b)
    t_$i = $stopwatch
    print t_$i
endloop
```

The above commands are stored in the sample script `DP_perf.inp`. The `verbose` string and the scalar 2 in `DPB_setoption` result in the printing of the values of parameters and gradients at each iteration in the log-likelihood maximisation. In addition, we use the option `vcv` with value 1 to set the covariance matrix estimator to OPG, which is the least computationally demanding. Since the `method` option was left unmodified, the above code estimates the dynamic probit by GHQ, with 16, 24, and 32 quadrature points. If, instead, one wishes to inspect the performance of the GHK algorithm, the method must be changed via `err = DPB_setoption(&b, "method", 1)` after the set-up function; in Table 3 we display the results obtained by setting the number of to 128, 192, and 256. The estimates obtained via GHQ do no exhibit remarkable differences from each other. Conversely, the GHK algo-

Table 3: DPB performance with different numbers of quadrature points/GHK replications. Covariance matrix in (3)

	γ (se)	θ (se)	σ_α (se)	log-lik	N. of iter.	time mm.ss.ms
GHQ						
16	0.6159 (.0349)	1.1120 (.0742)	0.9992 (.0337)	-10771.962	28	00:36:12
24	0.6159 (.0350)	1.1122 (.0743)	0.9990 (.0337)	-10771.971	28	00:38:43
32	0.6159 (.0349)	1.1122 (.0743)	0.9990 (.0337)	-10771.969	28	00:44:27
GHK						
128	0.6210 (.0347)	1.0984 (.0727)	0.9986 (.0336)	-10770.737	26	02:52:55
192	0.6191 (.0348)	1.1062 (.0733)	0.9979 (.0336)	-10771.302	25	03:52:65
256	0.6185 (.0348)	1.1067 (.0736)	0.9980 (.0336)	-10771.322	25	05:04:85

Results obtained on a system with two Intel(R) Pentium(R) CPU G640 @ 2.80GHz processors.

rithm obviously takes a longer time to yield the results because of its nature. In particular, the cpu-time increases in a roughly linear way with the number of replications.

As described in Section 2.1, the ADP and GADP models can be formulated by setting the variance-covariance matrix of the error terms as in (4) and (5) respectively. Such structure does not allow the use of GHQ to evaluate the relevant normal integral and GHK has to be used instead. In order to illustrate how DPB handles these extensions, we use the simulated dataset in (10) with $\rho = 0.3, \tau = 1$ (stored in ADP_artdata.gdtb) and $\rho = 0.3, \tau = 0.6$ (stored in GADP_artdata.gdtb) to build (4) and (5), respectively. The ADP model can be estimated by setting the the first argument of DPB_setup to "ADP":

```
open ADP_artdata.gdtb

list X = const x
list Z = const x z

b = DPB_setup("ADP",y,X,Z)
DPB_estimate(&b)
DPB_printout(&b)
```

The above code is also stored in the script file ADP_example.inp. In this case, if the user tries to set GHQ as estimation method, a warning message is printed and the estimation proceeds with GHK using 128 Halton points. The above code also returns the following output:

```
AR(1) Dynamic Probit model
Dependent variable: y
Units: 4096 (observations: 24576)
Covariance matrix: Sandwich
Method: GHK with 128 Halton points

             coefficient   std. error     z     p-value
  -----------------------------------------------------
  y(-1)       0.604903     0.0811089    7.458   8.79e-14 ***
```

```
           coefficient   std. error     z       p-value
  -----------------------------------------------------
  const       0.483515     0.0560415     8.628   6.25e-18  ***
  x           0.512773     0.0182398    28.11    6.81e-174 ***


           coefficient   std. error     z      p-value
  -----------------------------------------------------
  const       1.03196      0.0759177    13.59    4.40e-42 ***
  x           0.993716     0.0550138    18.06    6.23e-73 ***
  z           1.03108      0.128254      8.039   9.03e-16 ***


           coefficient   std. error     z       p-value
  -----------------------------------------------------
  theta       1.25590      0.0990524    12.68    7.72e-37  ***
  sigma       1.01745      0.0373731    27.22    3.37e-163 ***
  rho         0.306027     0.0563038     5.435   5.47e-08  ***

  Log-likelihood     -10057.724      AIC      20131.449
           BIC      20196.325      HQC      20152.464

  Wald test = 790.336 (1 df); p-value = 6.81115e-174
```

Similarly, the GADP model can be estimated by setting the first value in the `DPB_setup` to `"GADP"` (the code is provided in `GADP_example.inp`):

```
open GADP_artdata.gdtb

list X = const x
list Z = const x z

b = DPB_setup("GADP",y,X,Z)
```

After loading data with $\rho = 0.3$ and $\tau = 0.6$, running the estimation functions produces:

```
Generalised AR(1) Dynamic Probit model
Dependent variable: y
Units: 4096 (observations: 24576)
Covariance matrix: Sandwich
Method: GHK with 128 Halton points

           coefficient   std. error     z     p-value
  -----------------------------------------------------
  y(-1)       0.555213     0.0632615    8.776   1.69e-18 ***


           coefficient   std. error     z      p-value
  -----------------------------------------------------
  const       0.523555     0.0469495    11.15   7.05e-29  ***
  x           0.502173     0.0162414    30.92   6.57e-210 ***


           coefficient   std. error     z      p-value
  -----------------------------------------------------
  const        1.09461     0.0852157    12.85    9.16e-38 ***
  x            1.02195     0.0632144    16.17    8.70e-59 ***
  z            1.10415     0.131548      8.393   4.72e-17 ***
```

```
           coefficient   std. error     z       p-value
  -----------------------------------------------------
  theta       1.23328      0.122465     10.07    7.46e-24  ***
  sigma       1.01042      0.0393478    25.68    2.00e-145 ***
  rho         0.338685     0.0463474     7.308   2.72e-13  ***
  tau         0.528361     0.129116      4.092   4.27e-05  ***

  Log-likelihood      -9999.321      AIC      20014.641
           BIC      20079.517      HQC      20035.657

  Wald test = 956.004 (1 df); p-value = 6.57193e-210
```

4.2 Quadratic Exponential model

We exemplify the FE estimator proposed in Bartolucci and Nigro (2010), by means of artificial data set, generated in a similar way as in (10):

$$\begin{aligned} y_{it} &= 1\{0.6\, y_{it-1} + 0.5\, x_{1it} + 0.5\, x_{2it} + \alpha_i + \varepsilon_{it} \geq 0\} \\ y_{i1} &= 1\{0.5\, x_{1i1} + 0.5\, x_{2i1} \alpha_i + \varepsilon_{i1} \geq 0\} \quad \text{for } i = 1, \ldots, 1024 \;\; t = 1, \ldots, 6 \end{aligned}$$

where α_i is generated as in (10), the regressors x_{1it} and x_{2it} are standard normal random variables and the error terms ε_{it} are logistically distributed with zero mean and variance $\pi^2/3$. The file `QE_gendata.inp` contains the Hansl code for generating the artificial data also stored in `QE_artdata.gdtb`. A simple script to estimate the quadratic exponential model described in 2.2 is:

```
set echo off
set messages off

include DPB.gfn
open QE_artdata.gdtb

setobs id time --panel-vars

list X = x1 x2 x3

b = DPB_setup("QE",y,X)
DPB_setoption(&b, "vcv", 2)
DPB_estimate(&b)
DPB_printout(&b)
```

The above commands are also stored in `QE_example.inp`. After the required panel structure is set and the list of explanatory variables has been created, the script calls the set-up function with the string `"QE"` as its first argument. For illustrative purposes, in the above example the covariance matrix estimator is set to the Hessian by setting the `vcv` option to 2. The script returns the following output:

```
Quadratic Exponential model
Dependent variable: y
Units: 821 (observations: 4105)
Total units: 1024 (total observations: 5120)
Covariance matrix: Hessian

             coefficient   std. error     z     p-value
  ----------------------------------------------------
  y(-1)       0.612351     0.0911535    6.718   1.84e-11 ***
```

```
           coefficient   std. error     z      p-value
  -----------------------------------------------------
  x1        0.406810     0.0695830     5.846   5.02e-09 ***
  x2        0.556566     0.0512341    10.86    1.73e-27 ***
  x3        0.527469     0.0341211    15.46    6.59e-54 ***

           coefficient   std. error     z     p-value
  ----------------------------------------------------
  const     0.345770     0.103625     3.337   0.0008   ***
  x1        0.569998     0.133328     4.275   1.91e-05 ***
  x2        0.374503     0.0955802    3.918   8.92e-05 ***
  x3        0.496962     0.0697174    7.128   1.02e-12 ***

  Log-likelihood      -1081.463      AIC       2178.926
           BIC         2229.486      HQC       2196.825

  Wald test = 492.54 (3 df); p-value = 1.97431e-106
```

Notice that, since only the units with $0 < y_{i+} < T$, DPB reports in the output headings the number of actual contributions to the log-likelihood. The output reports coefficients, standard errors and z-statistics for the parameters in (8): the coefficient associated with the lagged dependent variable γ, the parameters $\boldsymbol{\beta}_1$ for the explanatory variables and, finally, the parameters associated with the last observation, $\left[\mu, \boldsymbol{\beta}_2'\right]'$.

The handling of time dummies in the model specification deserves a special mention. For instance, let us discuss the case of a balanced dataset: in a panel of T periods, the QE model identifies $T - 3$ time effects, as two dummies are dropped for the initial and rank condition in the main equation; another one gets dropped as the observation at time T is handled separately in the model specification (see expression (8)) and it includes an intercept term. An example is given in the script file `QE_example.inp`.

5 Examples: the `union` dataset

In this section, we illustrate the DPB function package by means of two empirical applications. In the first exercise, we replicate the example proposed by Stewart (2006), used to present the software components to estimate RE dynamic probit models in Stata.

In the second, we show how to implement in Hansl the estimator proposed by Wooldridge (2005), how to compute Partial Effects, and compare it with the DP model available in DPB. As a side-product, we show how to implement the popular Correlated Random Effects (CRE) approaches (Mundlak, 1978; Chamberlain, 1980) in Gretl and within the DPB function package.

Both examples use data extracted from the U.S. National Longitudinal Survey of Youth on unionised workers, a very popular dataset often used as a benchmark for RE (dynamic) models for binary dependent variables with longitudinal data.

5.1 Example 1

In the following, we replicate the empirical example provided in Stewart (2006), where the `union` dataset is used to illustrate the Stata software component for the estimation of the

RE dynamic probit models of Heckman (1981b), DP, and Hyslop (1999), ADP. [6] Differently from DPB, the software provided by Stewart (2006) unsurprisingly does not allow for the implementation of the estimator proposed by Keane and Sauer (2009). The results reported in this section can be replicated using the script file `union_example1.inp` and the data file `union_full.gdtb`.

Using the variables given in the `union` dataset, a model is specified for the binary dependent variable at time t and for the initial condition as in Heckman (1981b):

$$\begin{aligned} union_{it} &= 1\{\gamma\, union_{it-1} + \beta_0 + \beta_1\, age + \beta_2\, grade_{it} + \beta_3\, south_{it} + \alpha_i + \varepsilon_{it} \geq 0\} \\ union_{i1} &= 1\{\pi_0 + \pi_1\, age_{i1} + \pi_2\, grade_{i1} + \pi_3\, south_{i1} + \pi_4\, not\_smsa + \theta\alpha_i + \varepsilon_{i1} \geq 0\} \end{aligned}$$

for $i = 1, \ldots, n$ and $t = 2, \ldots, T$, where *union* is the binary dependent variable, *age* is the age at time t, *grade* are the years of schooling and *south* is a dummy variable for living in the South. For the initial condition, *not_smsa*, living outside a standard metropolitan statistical area, is used as an exclusion restriction.

The dataset is an unbalanced panel which starts in 1970 and ends in 1988; a few years do not appear in the dataset, so several gaps are present. The number units is 4434 and the maximum time length $T = 12$, for a total of 26200 observations. Simple descriptive statistics on the dataset are given by the summary command, followed by the variable names:

```
? summary union age grade south not_smsa --simple

                    Mean        Minimum        Maximum      Std. Dev.
union            0.22179         0.0000         1.0000        0.41546
age               30.432         16.000         46.000         6.4891
grade             12.761         0.0000         18.000         2.4117
south            0.41302         0.0000         1.0000        0.49238
not_smsa         0.28370         0.0000         1.0000        0.45080
```

In order to replicate the example provided in Stewart (2006), the dataset needs to be sub-sampled: only the years from 1978 are kept and 1983 is dropped; in addition, the panel is balanced by keeping units that are present for six consecutive waves. The following code sub-samples the dataset in order to keep the portion used in Stewart (2006) and stores it, for convenience, in the Gretl data file `union.gdtb`:

```
series valid = 0

smpl year>=78 --restrict
smpl year!=83 --restrict

matrix m = aggregate(const,idcode)
series nwav = replace(idcode, m[,1], m[,2])
smpl nwav==6 --restrict
series valid = 1

store union.gdtb
open union.gdtb --quiet

smpl full
smpl valid --dummy
setobs idcode year --panel-vars
```

[6] The data for this example were retrieved from the URL . As far as we are aware, usage of this dataset was unrestricted for academic use and we believe that a comparison between our results and Stewart's was indispensable.

The resulting dataset is a balanced panel of 799 units observed for $T = 6$ periods, for a total of 3995 observations.

The models estimated in Stewart (2006) are a pooled probit, a RE probit (with no special treatment for the initial conditions), the DP and the ADP. The first two columns of Table 4 show the estimation results of the pooled probit and of the the RE probit. In the first case, the estimates of both the coefficients and standard errors are identical to those obtained by Stewart (2006), whereas the estimates of the RE probit (with 24 quadrature points) exhibit very small differences (the coefficient associated with $union_{t-1}$ is 1.1507 in Stewart (2006) compared to 1.1509, and so on).

The third column of Table 4 reports the estimation results of the DP model, using 24 quadrature points. In this case, estimated coefficients, standard errors and the value of the log-likelihood at convergence are identical.[7] The fourth and fifth columns of Table 4 show the estimation results of the DP model using GHK instead of GHQ and of the ADP, both using 500 Halton points. While substantially similar, the results do present some discrepancies from those obtained by Stewart (2006). Such differences can probably be ascribed the use of Halton sequences in `DPB` as opposed to random draws form the uniform distribution in the command written by Stewart (2006). Nevertheless, both sets of estimates show that ignoring the autocorrelation in the unobservables leads to the underestimation of the state dependence effect in these data. Finally, Table 5 reports the estimation results for the GADP and the QE model. In the first case, the results are very similar results to those obtained by estimating the ADP model, which is expected since the additional correlation coefficient τ is positive and close to unity. In the second, the values of the estimated coefficients are coherent with those of the RE models with no autocorrelation.

Since `DPB` handles unbalanced datasets, we repeat the exercise keeping all the usable observations. Starting from the full dataset at our disposal (`union_full.gdtb`), the `DPB_setup` function for RE models automatically sub-samples the dataset as described in Section 3: out of the 4434 units 3790 are kept, while 400 are dropped because observed only once and 244 because are never observed for at least two consecutive periods. Table 6 reports the estimation results based on this dataset and only for the models implemented in `DPB`. Retaining all usable units in the dataset considerably increases the estimate of the state dependence parameter in all models while reduces the estimated persistence due to time invariant unobserved heterogeneity.

5.2 Example 2

In the following, we show how to implement in Gretl the popular RE estimator for dynamic binary models proposed by Wooldridge (2005), CDP (*Conditional DP*) henceforth. To this aim, we use the same dataset available in the data archive of the Journal of Applied Econometrics, also available in Gretl data archive. The data are a balanced panel of workers, extracted from the U.S Longitudinal Survey of Youth, which comprises 545 observed for $T = 7$ periods, from 1981 to 1987.

The method proposed by Wooldridge (2005) relies on specifying a distribution for the individual unobserved heterogeneity conditional on the initial value of the dependent variable and the observed history of strictly exogenous explanatory variables. Following the paper, to which we refer the reader to for details, the distribution of the individual unobserved effect is specified as

[7] The reader should keep in mind that we directly estimate σ_α while Stewart (2006) estimates $\lambda = \frac{\sigma_\alpha^2}{\sigma_\alpha^2+1}$.

Table 4: Estimation results: Replication of Stewart (2006), Table 1, p. 265

	Pooled probit	RE probit (static)	DP	DP with GHK	ADP
union(-1)	1.8849 (.0525)	1.1509 (.1419)	0.6344 (.0983)	0.6343 (.1012)	1.3181 (.1554)
const	-0.6986 (.2474)	0.1785 (.4181)	0.5633 (.4798)	0.5956 (.4818)	0.0948 (.4034)
age	-0.0087 (.0058)	-0.0240 (.0085)	-0.0286 (.0092)	-0.0294 (.0092)	-0.0237 (.0081)
grade	-0.0145 (.0103)	-0.0386 (.0207)	-0.0539 (.0269)	-0.0539 (.0269)	-0.0369 (.0201)
south	-0.1684 (.0519)	-0.3691 (.1034)	-0.4883 (.1239)	-0.4953 (.1247)	-0.3742 (.0998)
const			-0.9597 (.8414)	-0.9419 (.8417)	-0.8710 (.8496)
age			0.0081 (.0238)	0.0077 (.0238)	0.0100 (.0243)
grade			-0.0064 (.0341)	-0.0064 (.0341)	-0.0132 (.0334)
south			-0.7261 (.1651)	-0.7310 (.1650)	-0.7559 (.1670)
not smsa			-0.4152 (.1644)	-0.4151 (.1644)	-0.4181 (.1662)
θ			0.8641 (.1095)	0.8622 (.1106)	1.2250 (.2143)
σ			1.5261 (.1251)	1.5293 (.1619)	1.0438 (.1519)
ρ					- 0.3372 (.0554)
$\ln(\sigma^2)$		0.0909 (.2923)			
Log-lik	-1573.642	-1563.184	-1860.215	-1860.275	-1854.618

Standard errors (in parentheses) are computed using the Hessian. Number of quadrature points: 24; Number of GHK Halton draws: 500.

$$\alpha_i | y_{i1}, \boldsymbol{w}_i \sim N(\delta_0 + \delta_1 y_{i1} + \delta_2' \boldsymbol{w}_i ; s_\alpha^2)$$

where y_{i1} is the initial observation and $\boldsymbol{w}_i$ contains the whole sequence of the strictly exogenous covariate w_{it} as in the approach of Chamberlain (1980). Notice that we use a different notation of the variance parameter as we will use σ_α^2 to denote the variance of the unconditional distribution.

5.2.1 Estimation of the CDP probit

As is well known, the main virtue of the CDP model is that it can be easily estimated by means of standard routines used for the static RE probit with additional regressors. In Section 6 of Wooldridge (2005) the CDP model is estimated with $union_{it}$ as the dependent variable, a dummy variable $married_{it}$ and time dummies as explanatory variables. In addition, the whole history of *married* and the initial observation $union_{i80}$ are included in the set of regressors to account for the conditional distribution of α_i. A second specification is also used, where the set of regressors contains also *educ* (years of schooling) and the dummy *black*. The practice of including lags and leads of a time-varying strictly exogenous variable is often adopted by practitioners as a preventive measure against possible correlation between the random effect and the covariates. For this reason, a static/dynamic model with this feature is often referred to as a CRE (Correlated Random Effects) approach.

The Hansl script to replicate the results in Table I, p. 52 of Wooldridge (2005) is

Table 5: Estimation results: GADP and QE models

	GADP	QE
union(-1)	1.3006 (.1618)	1.0638 (.17060)
const	0.1164 (.4101)	
age	-0.0241 (.0082)	-0.0889 (.0231)
grade	-0.0376 (.0204)	-0.1907 (.1811)
south	-0.3791 (.1016)	-0.1187 (.4895)
const	-0.8824 (.8421)	0.5677 (.9465)
age	0.0101 (.0241)	-0.1140 (.0516)
grade	-0.0121 (.0332)	-0.1105 (.1850)
south	-0.7486 (.1659)	0.0266 (.5355)
not smsa	-0.4149 (.1648)	
θ	1.1813 (.2256)	
σ	1.0635 (.1619)	
ρ	-0.3419 (.0567)	
τ	0.8499 (.1877)	
$\ln(\sigma^2)$		
Log-lik	-1854.534	-467.495

Standard errors (in parentheses) are computed using the Hessian. Number of GHK Halton draws: 500. For the last column, the coefficients related to the second block of covariates refers to the parameters for the last observation in (7).

```
list TIME = dummify(year)
TIME -= Dyear_2
list MARR = marr8*
list X = const union_1 union80 married MARR TIME

probit union X --random-effects --quadpoints=12

probit union X educ black --random-effects --quadpoints=12
```

where, after creating the list of explanatory variables, the two `probit` commands with the `--random-effects` flag estimate the CDP models. The default number of quadrature points has to be set to 12 to have an exact replication of Wooldridge's original results.

For comparison purposes, we use the same specifications to estimate the DP model with the CRE approach. The code, also containing the above script, is given in the file `union_example21.inp` and the estimation results are displayed in Table 7. The results in the first two columns of the table perfectly replicate the results in Wooldridge (2005). In addition, the same models estimated as DP models produce very similar results, as to be expected from the simulation study in Akay (2012).

The Mundlak version (Mundlak, 1978) of the CRE approach is also very popular among practitioners: instead of lags and leads, the regression is augmented by the within group mean of one or more time-varying strictly exogenous covariate. This estimator can easily

Table 6: Estimation results: DP, ADP, GADP and QE models, 3790 units

	DP	ADP	GADP	QE
union(-1)	0.9183 (.0472)	1.5168 (.0636)	1.5129 (.0648)	1.6640 (.0735)
const	-1.7983 (.1749)	-1.6159 (.1402)	-1.6185 (.1408)	
age	0.0029 (.0033)	0.0005 (.0028)	0.0005 (.0028)	0.0143 (.0060)
grade	0.0316 (.0110)	0.0200 (.0087)	0.0201 (.0087)	-0.0802 (.0577)
south	-0.4206 (.0530)	-0.3215 (.0434)	-0.3225 (.0437)	-0.6180 (.1665)
const	-1.8242 (.2238)	-1.6905 (.2159)	-1.6857 (.2156)	-0.4120 (.6163)
age	-0.0011 (.0052)	-0.0052 (.0051)	-0.0051 (.0051)	0.0372 (.0146)
grade	0.0659 (.0136)	0.0682 (.0130)	0.0678 (.0131)	-0.0782 (.0590)
south	-0.4674 (.0683)	-0.4741 (.0665)	-0.4721 (.0666)	-0.6912 (.1956)
not smsa	0.0031 (.0691)	-0.0174 (.0686)	-0.0170 (.0684)	
θ	0.8405 (.0575)	1.0855 (.0908)	1.0714 (.0996)	
σ	1.1735 (.0535)	0.8305 (.0508)	0.8348 (.0524)	
ρ		-0.3207 (.0246)	-0.3233 (.0258)	
τ			0.9461 (.0766)	
Log-lik	-7480.237	-7445.438	-7445.387	-2348.336

Standard errors (in parentheses) are computed using the Hessian. Number of quadrature points: 24; Number of GHK Halton draws: 500.

be implemented in Gretl by means of dedicated panel-data functions, available to the user after setting the panel data structure. In the script file `union_example21.inp`, the within group mean of the variable *married* is readily created in the line `m_marr = pmean(married)`. Using the baseline specification in Wooldridge (2005), we estimate the CDP and DP models with the Mundlak's CRE specification. The results are displayed in Table 8.

5.2.2 Computation of Partial Effects

One of the reasons why RE models are often preferred to FE models by practitioners is the possibility of computing Partial Effects (PE), that have a meaningful economic interpretation in terms of probability variations. While the computation of PE is straightforward in models for cross-sectional data, additional sources of complication arise when dealing with panel data, mainly because the individual unobserved effect α_i enters the index function. However, in CRE models, the unobserved heterogeneity is represented by the estimated conditional mean of α_i, that enters the predicted probability additively to the index function. In the following we show how to compute predicted probabilities by Hansl scripting, that can be used to derive PE or Average PE (APE).

As an example, Wooldridge (2005) computes the estimated probability of being in a union in 1987 conditional on being or not in a union in 1986 and on being married or not. After estimating the RE model and saving the relevant quantities, the predicted probabilities can easily be computed by means of a few lines of code. For instance, the probability of being in a union in 1987, conditional on being in a union in 1986 and being married,

$\Pr(union_{i1987} = 1 \mid union_{i1986} = 1,\ married_{it} = 1,\ \boldsymbol{x}_{it}, \boldsymbol{w}_i)$, can written as

$$\Phi\left(\frac{\hat{\gamma} + \hat{\delta}_0 + \hat{\delta}_1 union_{i1980} + \hat{\beta} + \boldsymbol{w}_i'\hat{\delta}_2 + \hat{\beta}_{1987}}{\sqrt{1+s_\alpha^2}}\right)$$

and is computed as

```
list X = const union_1 union80 married MARR TIME

probit union  X --random-effects --quadpoints=12 --quiet

k   = nelem(X)
bw  = $coeff[1:k]
s2a = exp($coeff[k+1])

series ndx = bw[1] + bw[2] + bw[3]*union80 + bw[4] + lincomb(MARR,bw[5:11]) + bw[17]
series ndx /= sqrt(1+s2a)
wp11_m = cnorm(ndx)
```

After the model estimation command is invoked on the second line, we retrieve the estimated parameters by means of the accessor `$coeff`, which contains the parameter $\ln(s_\alpha)$ at the end. Then, we proceed to the computation of the index function, where `bw[1]` is $\hat{\delta}_0$, `bw[2]` is $\hat{\gamma}$, `bw[3]` is $\hat{\delta}_1$, `bw[4]` is the coefficient that multiplies the dummy *married*, `lincomb(MARR,bw[5:11])` is $\boldsymbol{w}_i'\hat{\delta}_2$ and `bw[17]` is the coefficient for the 1987 time dummy. After the index function is normalised by $\sqrt{1+\hat{s}_\alpha^2}$, the predicted probability is computed by the function `cnorm`, which returns the standard normal cdf. Similarly the estimated probabilities $\Pr(union_{i1987} = 1 \mid union_{i1986} = 0,\ married_{it} = 1,\ \boldsymbol{x}_{it}, \boldsymbol{w}_i)$, and for $married_{it} = 0$, can be computed by appropriately excluding `bw[2]` and `bw[4]` from the index function calculation.

The top panel of Table 9 reports the average estimated probabilities in the four cases and is the equivalent of Table II on page 52 in Wooldridge (2005). APEs can readily be derived by taking cell differences: for instance, the state dependence for married individuals is 0.182, and so on. The lower panel of Table 9 reports the results for the same exercise performed on the estimated coefficients by the Heckman's RE dynamic probit. In this case, the script file looks like

```
list X = const married MARR TIME
list Z = const married

b = DPB_setup(1,union,X,Z)
DPB_setoption(&b, "nrep", 12)
DPB_setoption(&b, "verbose", 0)
DPB_estimate(&b)

k = b.nk
z = b.nz
bh = b.coeff[1:k+1]
sig_a = b.coeff[k+z+3]
s2a = sig_a^2

series ndx = bh[1] + bh[2] + bh[3] + lincomb(MARR, bh[4:10]) + bh[16]
series ndx =/sqrt(1+s2a)
series hp11_m = cnorm(ndx)
```

This time, after estimating the model, estimated coefficients needs to be extracted form the bundle `b`, where they are stored into the vector `coeff`. The syntax `bh = b.coeff[1:k+1]` and `sig_a = b.coeff[k+z+3]` extracts the needed estimates according to the order in which they are stored. The index function and the estimated probabilities are computed as in the previous exercise.

The script file `union_example22.inp` contains the code to replicate the full Table 9. An alternative way to compute estimated probabilities would be to use the Gauss-Hermite quadrature to integrate α_i out of the standard normal cdf. For brevity we do not illustrate this procedure here, however we do provide the script file `union_example22_quad.inp` that computes the same average estimated probabilities by numerical integration.

6 Conclusions

The aim of the DPB function package is to provide the practitioner with an intuitive and simple-to-use tool for the estimation of panel data dynamic binary choice models, whose adoption is often called for in applied microeconometrics. Hopefully, DPB will allow practitioners, who may otherwise be discouraged by the complexity that arises from implementing these estimators in-house, to easily employ these estimators in standard research problems.

Acknowledgements

We are immensely grateful to Matteo Picchio, for thoroughly testing the function package and for his many comments on the manuscript, and to Allin Cottrell for providing assistance with the Gretl supporting infrastructure.

Table 7: Estimation results: Replication of Wooldridge (2005), Table I, p. 52 and DP models, Chamberlain's CRE approach

	CDP (1)	CDP (2)	DP (1)	DP (2)
union(-1)	0.8747 (.0944)	0.8857 (.0942)	0.8866 (.1090)	0.8885 (.1128)
const	-1.8276 (.1522)	-1.7124 (.4493)	-1.4907 (.1373)	-1.4921 (.5033)
married	0.1676 (.1110)	0.1691 (.1110)	0.1686 (.1120)	0.1703 (.1126)
year82	0.0280 (.1141)	0.0274 (.1139)	0.0303 (.1276)	0.0297 (.1276)
year83	-0.0880 (.1179)	-0.0893 (.1178)	-0.0836 (.1125)	-0.0869 (.1126)
year84	-0.0484 (.1195)	-0.0508 (.1194)	-0.0399 (.1196)	-0.0429 (.1197)
year85	-0.2675 (.1230)	-0.2681 (.1228)	-0.2587 (.1211)	-0.2591 (.1215)
year86	-0.3190 (.1250)	-0.3173 (.1248)	-0.3085 (.1212)	-0.3109 (.1216)
year87	0.0738 (.1194)	0.0727 (.1192)	0.0789 (.1292)	0.0772 (.1296)
educ		-0.0169 (.0361)		-0.0126 (.0426)
black		0.5349 (.1942)		0.7816 (.2055)
marr81	0.0637 (.2088)	0.0546 (.2071)	0.1173 (.1829)	0.1136 (.2007)
marr82	-0.0707 (.2556)	-0.0606 (.2458)	-0.0752 (.2467)	0.0216 (.2908)
marr83	-0.1292 (.2425)	-0.1363 (.2423)	-0.0824 (.2453)	-0.1403 (.2956)
marr84	0.0251 (.2651)	0.0698 (.2678)	-0.0019 (.2610)	-0.0330 (.2463)
marr85	0.4070 (.2459)	0.4282 (.2445)	0.3621 (.2235)	0.3367 (.1801)
marr86	0.1089 (.2626)	0.0789 (.2628)	0.1915 (.2489)	0.2394 (.2112)
marr87	-0.4266 (.2106)	-0.3878 (.2157)	-0.5065 (.1907)	-0.4085 (.1657)
union80	1.5144 (.1646)	1.4771 (.1706)		
const			-0.9775 (.1013)	-0.6868 (.5808)
married			0.2279 (.1907)	0.2462 (.1886)
educ				-0.0291 (.0486)
black				0.5401 (.2363)
θ			0.7134 (.0993)	0.6991 (.1040)
σ_α			1.3181 (.1083)	1.3097 (.1307)
$ln(s_\alpha)$	0.2435 (.1812)	0.1885 (.1788)		
Log-lik	-1287.475	-1283.390	-1594.371	-1587.094

Table 8: Estimation results: CDP and DP models, Mundlak's CRE approach

	CDP	DP
union(-1)	0.8875 (.0925)	0.8988 (.1068)
const	-1.8568 (.1420)	-1.5462 (.1362)
married	0.1698 (.1103)	0.1677 (.1109)
year82	0.0293 (.1138)	0.0299 (.1271)
year83	-0.0889 (.1177)	-0.0865 (.1117)
year84	-0.0494 (.1192)	-0.0402 (.1191)
year85	-0.2673 (.1227)	-0.2579 (.1205)
year86	-0.3182 (.1246)	-0.3095 (.1207)
year87	0.0717 (.1192)	0.0762 (.1289)
m_marr	0.0332 (.1993)	0.0570 (.2035)
union80	1.4776 (.1630)	
const		-0.9569 (.1005)
married		0.1956 (.1805)
θ		0.6962 (.0941)
σ_α		1.3058 (.1040)
$ln(s_\alpha)$	0.2295 (.1694)	
Log-lik	-1291.377	-1598.478

Table 9: Probability of being in a union in 1987

	Replication of Wooldridge (2005), Table II, p.52	
	In union, 1986	Not in union, 1986
Married, 1987	0.4082	0.2256
Not married, 1987	0.3696	0.1970
	DP	
	In union, 1986	Not in union, 1986
Married, 1987	0.4081	0.2211
Not married, 1987	0.3680	0.1922

References

Akay, A. (2012). Finite-sample comparison of alternative methods for estimating dynamic panel data models. *Journal of Applied Econometrics*, 27(7):1189–1204.

Albarrán, P., Carrasco, R., and Carro, J. (2015). Estimation of dynamic nonlinear random effects models with unbalanced panels. Economics working papers, Universidad Carlos III, Departamento de Economìa.

Alessie, R., Hochguertel, S., and van Soest, A. (2004). Ownership of Stocks and Mutual Funds: A Panel Data Analysis. *The Review of Economics and Statistics*, 86(3):783–796.

Arulampalam, W. (2002). State dependence in unemployment incidence: evidence for british men revisited. Technical report, IZA Discussion paper series.

Arulampalam, W. and Stewart, M. B. (2009). Simplified implementation of the heckman estimator of the dynamic probit model and a comparison with alternative estimators*. *Oxford bulletin of economics and statistics*, 71(5):659–681.

Bartolucci, F., Bellio, R., Sartori, N., and Salvan, A. (2014). Modified profile likelihood for fixed-effects panel data models. *Econometric Reviews*, in press.

Bartolucci, F. and Nigro, V. (2010). A dynamic model for binary panel data with unobserved heterogeneity admitting a $\sqrt{n}$-consistent conditional estimator. *Econometrica*, 78:719–733.

Bettin, G. and Lucchetti, R. (2012). Intertemporal remittance behaviour by immigrants in germany. Working Papers 385, Universita' Politecnica delle Marche (I), Dipartimento di Scienze Economiche e Sociali.

Biewen, M. (2009). Measuring state dependence in individual poverty histories when there is feedback to employment status and household composition. *Journal of Applied Econometrics*, 24(7):1095–1116.

Brown, S., Ghosh, P., and Taylor, K. (2012). The existence and persistence of household financial hardship. Working papers, The University of Sheffield, Department of Economics.

Butler, J. S. and Moffitt, R. (1982). A computationally efficient quadrature procedure for the one-factor multinomial probit model. *Econometrica*, 50(3):761–64.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16:1190–1208.

Cappellari, L. and Jenkins, S. P. (2004). Modelling low income transitions. *Journal of Applied Econometrics*, 19(5):593–610.

Carrasco, R. (2001). Binary choice with binary endogenous regressors in panel data. *Journal of business & economic statistics*, 19(4).

Carro, J. (2007). Estimating dynamic panel data discrete choice models with fixed effects. *Journal of Econometrics*, 140:503–528.

Carro, J. M. and Traferri, A. (2012). State dependence and heterogeneity in health using a bias-corrected fixed-effects estimator. *Journal of Applied Econometrics*.

Chamberlain, G. (1980). Analysis of covariance with qualitative data. *The Review of Economic Studies*, pages 225–238.

Chamberlain, G. (1985). Heterogeneity, omitted variable bias, and duration dependence. In Heckman, J. J. and Singer, B., editors, *Longitudinal analysis of labor market data*. Cambridge University Press: Cambridge.

Contoyannis, P., Jones, A. M., and Rice, N. (2004). Simulation-based inference in dynamic panel probit models: an application to health. *Empirical Economics*, 29(1):49–77.

Fernández-Val, I. (2009). Fixed effects estimation of structural parameters and marginal effects in panel probit models. *Journal of Econometrics*, 150(1):71–85.

Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–149.

Geweke, J. (1989). Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 57(6):1317–1339.

Giarda, E. (2013). Persistency of financial distress amongst italian households: Evidence from dynamic models for binary panel data. *Journal of Banking & Finance*, 37(9):3425 – 3434.

Hahn, J. and Kuersteiner, G. (2011). Bias reduction for dynamic nonlinear panel models with fixed effects. *Econometric Theory*, 27(06):1152–1191.

Hahn, J. and Newey, W. (2004). Jackknife and analytical bias reduction for nonlinear panel models. *Econometrica*, 72:1295–1319.

Hajivassiliou, V. A. and McFadden, D. L. (1998). The method of simulated scores for the estimation of ldv models. *Econometrica*, 66(4):863–896.

Halliday, T. J. (2008). Heterogeneity, state dependence and health. *The Econometrics Journal*, 11(3):499–516.

Heckman, J. J. (1981a). Heterogeneity and state dependence. *Structural Analysis of Discrete Data with Econometric Applications*, MIT Press: Cambridge MA. Manski CF, McFadden (eds).

Heckman, J. J. (1981b). The incidental parameters problem and the problem of initial conditions in estimating a discrete time-discreted data stochastic process. *Structural Analysis of Discrete Data with Econometric Applications*, MIT Press: Cambridge MA. Manski CF, McFadden (eds).

Heckman, J. J. and Borjas, G. J. (1980). Does unemployment cause future unemployment? definitions, questions and answers from a continuous time model of heterogeneity and state dependence. *Economica*, 47(187):pp. 247–283.

Heiss, F. (2011). Dynamics of self-rated health and selective mortality. *Empirical economics*, 40(1):119–140.

Honoré, B. E. and Kyriazidou, E. (2000). Panel data discrete choice models with lagged dependent variables. *Econometrica*, 68:839–874.

Hyslop, D. R. (1999). State dependence, serial correlation and heterogeneity in intertemporal labor force participation of married women. *Econometrica*, 67(6):1255–1294.

Keane, M. P. (1994). A computationally practical simulation estimator for panel data. *Econometrica*, 62:95–116.

Keane, M. P. and Sauer, R. M. (2009). Classification error in dynamic discrete choice models: Implications for female labor supply behavior. *Econometrica*, 77(3):975–991.

Krailo, M. D. and Pike, M. C. (1984). Algorithm as 196: Conditional multivariate logistic analysis of stratified case-control studies. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 33(1):95–103.

Miranda, A. (2007). Dynamic probit models for panel data: A comparison of three methods of estimation. In *United Kingdom Stata Users' Group Meetings*, volume 11.

Mundlak, Y. (1978). On the Pooling of Time Series and Cross Section Data. *Econometrica*, 46(1):69–85.

Orme, C. D. (1997). The initial conditions problem and two-step estimation in discrete panel data models. mimeo, University of Manchester.

Orme, C. D. (2001). Two-step inference in dynamic non-linear panel data models. mimeo, University of Manchester.

Pigini, C., Presbitero, A. F., and Zazzaro, A. (2014). State dependence in access to credit. Mo.Fi.R. Working Papers 102, Money and Finance Research group (Mo.Fi.R.) - Univ. Politecnica Marche - Dept. Economic and Social Sciences.

Rabe-Hesketh, S., Skrondal, A., and Pickles, A. (2002). Reliable estimation of generalized linear mixed models using adaptive quadrature. *Stata Journal*, 2(1):1–21.

Saito, M. and Matsumoto, M. (2008). SIMD-oriented Fast Mersenne Twister: a 128-bit pseudorandom number generator. In Keller, A., Heinrich, S., and Niederreiter, H., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 607–622. Springer, Berlin.

Stewart, M. (2006). Maximum simulated likelihood estimation of random-effects dynamic probit models with autocorrelated errors. *Stata Journal*, 6(2):256–272.

Stewart, M. B. (2007). The interrelated dynamics of unemployment and low-wage employment. *Journal of applied econometrics*, 22(3):511–531.

Wooldridge, J. M. (2005). Simple solutions to the initial conditions problem in dynamic, nonlinear panel data models with unobserved heterogeneity. *Journal of applied econometrics*, 20(1):39–54.

Wunder, C. and Riphahn, R. T. (2014). The dynamics of welfare entry and exit amongst natives and immigrants. *Oxford Economic Papers*, 66(2):580–604.