THE UNIVERSITY of York

Discussion Papers in Economics

No. 2002/09

Data Augmentation in Limited-Dependent Variable Models

by

Roberto Leon-Gonzalez

Department of Economics and Related Studies
University of York
Heslington
York, YO10 5DD

Data Augmentation in Limited-Dependent Variable Models.

June 2002.

Roberto Leon-Gonzalez.
Centre for Health Economics
& Department of Economics
University of York,
YO10 5DD, U.K.
email: rlg103@york.ac.uk

Abstract: This paper proposes a scheme that speeds up the convergence of Markov Chain Monte Carlo (MCMC) algorithms in the context of limited-dependent variable models. The algorithm reduces autocorrelations more than the recently proposed Parameter Expansion Data Augmentation (PX-DA) algorithm. In addition, the paper provides an algorithm to sample a variance-covariance matrix with restrictions directly from the conditional posterior distribution. Finally, it is shown that the PX-DA algorithm, as applied to the multivariate probit model, can be seen as sampling from a different parameterization of the model. However, in some cases the PX-DA algorithm is not invariant to reparameterizations, and a slightly different algorithm is proposed.

Key Words: Data Augmentation, Parameter-Expansion-Data-Augmentation, Inverted Wishart, Multivariate Probit, Reparameterization.

1 Introduction.

The rapid development of Markov Chain Monte Carlo (MCMC) techniques during the last decade has made possible the Bayesian analysis of models with complex likelihoods. The pioneer works of Metropolis et al. (1953) and Hastings (1970), were complemented with the ideas of data augmentation in Tanner and Wong (1988). Data augmentation consists in regarding latent and missing data as parameters to estimate. Although this introduces many more parameters, the conditional distributions became much easier to sample from.

Although data augmentation makes it easy to design an algorithm, convergence is slow due to the high correlation between model parameters and latent data. Hence, the chain moves slowly along the parameter space. Slow movement is not only a problem of time, but more importantly, it makes it more difficult to determine when the chain has reached convergence. Even when it seems plausible that a slow chain has converged, it will need an extraordinary large number of iterations to recover the parameter space enough times to get a representative sample from the posterior distribution. Hence the advise (e.g. Raftery and Lewis 1992, Gilks and Roberts 1995) that when a chain is very slow an alternative algorithm must be designed.

It is well known that in simple models, standard algorithms such as the Gibbs sampler work very well. However, as Gilks and Roberts (1995) note, MCMC algorithms applied to more ambitious models may perform poorly, and new strategies must be explored. The aim of this paper is to provide the practitioner of MCMC with novel tools to reduce the autocorrelations of the chain. By reducing autocorrelations, the convergence pattern of the chain can be detected more easily, hence enhancing the reliability of the calculations.

To put things in context, this paper will focus on a multivariate probit, although the techniques can be applied to a wide range of limited-dependent variable models. Let Y_i be a vector of zeros and ones. In the multivariate probit model, each component y_{it} of Y_i is determined by a continuous unobserved latent variable y_{it}^* generated according to the following process,

$$y_{it}^* = X_{it}\beta_t + e_{it} \qquad i = 1, ..., N \qquad t = 1, ..., T \tag{1}$$

the vector $e_i = (e_{i1}, ..., e_{iT})^T$ is normally distributed with zero mean and covariance matrix $\Sigma = (\sigma_{jk})$. The binary variable y_{it} is equal to one if and only if $y_{it}^* \geq 0$, and is equal to zero otherwise. X_{it} is a $1 \times k_t$ vector of regressors and β_t is a vector of parameters.

The posterior distribution of this model $\pi_M(\beta_1, ..., \beta_T, \Sigma | Y_1, ..., Y_N)$ does not belong to any standard family of distributions, and hence its characteristics such as mean and variance are not known analitically. In the case of $T \leq 3$, $\pi_M(\beta_1, ..., \beta_T, \Sigma | Y_1, ..., Y_N)$ can be evaluated numerically, and hence a Metropolis algorithm to sample from the posterior is in theory possible. However, a Metropolis algorithm requires finding a distribution that approximates well the posterior, and this is a difficult task when there are many parameters in the model.

Another approach is to analyse the posterior distribution in the augmented model, $\pi_{DA}(\beta_1, ..., \beta_T, \Sigma, Y_1^*, ..., Y_N^* | Y_1, ..., Y_N)$. This is the approach followed by Chib and Greenberg (1998). Although this distribution has many more variables, the conditional distributions belong to well known families and therefore it is possible to apply a Gibbs sampling algorithm (Gelfand and Smith 1990). The Gibbs algorithm divides the parameters into several groups and samples from a distribution by iteratively sampling from the conditional distribution of the parameters in one group given the parameters in the rest of groups. However, the larger the number of groups is, the slower the convergence (e.g. Gilks et al. 1995, page 12). And also, the more correlated the parameters in one group are with the parameters in another group, the slower the algorithm will be (e.g. Amit 1991, Roberts 1995, page52).

A recent strand of literature is concerned with the acceleration of data augmentation techniques (e.g. Meng and van Diyk 1997, Liu and Wu 1998, Liu and Sabatti 1998, Liu and Sabatti 2000, van Diyk and Meng 2001). They propose to introduce non-identified parameters into the model. The introduction of such parameters proves useful to speed up convergence.

The algorithm proposed in section 4 of this paper combines both the Gibbs sampling and the Metropolis algorithm. That is, data augmentation is used, but the parameters are also updated marginally on the latent data. The parameters are updated using a re-parameterization of posterior $\pi_{M(.)}$. As it is shown in section 3, the algorithm proposed by Liu and Wu (1998) can be seen as sampling from a re-parameterization of $\pi_{DA(.)}$. Simulations show that the proposed algorithm moves faster. This accords with the theory, since $\pi_{M(.)}$ conditions on less parameters.

A common feature in limited-dependent variable models is that the scale of the latent variable is not identified. As a normalization, restrictions are usually placed upon the variance-covariance matrix. When just one of the elements in the variance-covariance matrix is restricted to be one, several algorithms to sample it directly are available (Cowles 1996, McCulloch et al. 2000 and Nobile 2000). In the context of the multivariate probit model, where all the diagonal elements of the covariance matrix are restricted, Lui (2001) proposes an algorithm to sample the covariance matrix directly. However, the algorithm relies on the specification of an improper prior for the covariance matrix. There are no results that ensure that the posterior density would exist if an improper prior is specified, and hence the specification of a proper prior seems to be necessary. By choosing an alternative normalization, section 2 proposes an algorithm to sample Σ directly from its conditional posterior distribution using a proper prior.

The plan of the paper is as follows. Section 2 explains how to sample Σ directly in a Gibbs sampling algorithm. Section 3.1 shows that the Parameter-Expansion-Data-Augmentation (PX-DA) algorithm (Liu and Wu 1999), as applied to the univariate probit, can be seen as sampling from a reparameterization of the model. Section 3.2 adapts the PX-DA algorithm to the Multivariate Probit model with proper priors. Section 3.3 determines under which conditions the PX-DA algorithm can be seen in general as a reparameterization. And it illustrates that the PX-DA algorithm is not invariant to reparameterizations.

Section 4 describes an algorithm that reduces autocorrelations further, and that updates the absolute value of the parameters marginally on the latent data. Section 5 considers an algorithm that updates the absolute value of parameters and latent data jointly. Section 6 presents a Metropolis algorithm to sample from $\pi_{M(.)}$, using an approximation of the posterior that works well with a moderate number of parameters. Section 7 compares the proposed algorithms with other algorithms in the literature. Section 8 concludes.

2 Sampling a Variance-Covariance with Restrictions.

For simplicity in the exposition, the next sub-section concentrates on the case of T=2, known as the bivariate probit model. Section 2.2 looks at the more general case and section 2.3 considers a different type of normalization.

2.1 The Bivariate Probit Case.

2.1.1 Identification in the Bivariate Probit.

The likelihood contribution of an observation $(0,1)$ is,

$$\Pr\{y_{i1}=0, y_{i2}=1\} = \Pr\left\{ \begin{array}{c} X_{i1}\beta_1 + e_{i1} \leq 0 \\ X_{i2}\beta_2 + e_{i2} > 0 \end{array} \right\} \tag{2}$$

Let $\Delta = (\delta_{jk})$ be the lower triangular Cholesky decomposition of Σ, so that $\Sigma = \Delta\Delta^T$.

$$\Delta = \left(\begin{array}{cc} \sqrt{\sigma_{11}} & 0 \\ \sigma_{12}/\sqrt{\sigma_{11}} & \sqrt{\sigma_{22} - (\sigma_{12})^2/\sigma_{11}} \end{array} \right) \tag{3}$$

Then the vector e_i can be seen as a transformation of a random vector ε_i that follows a standard normal distribution. That is, $e_i = \Delta\varepsilon_i$, where ε_i follows a $N(0,I)$. The probability in (2) can be rewritten as,

$$\Pr\{y_{i1}=0, y_{i2}=1\} = \Pr\left\{ \begin{array}{c} X_{i1}\beta_1 + \sqrt{\sigma_{11}}\varepsilon_{i1} \leq 0, \\ X_{i2}\beta_2 + \frac{\sigma_{12}}{\sqrt{\sigma_{11}}}\varepsilon_{i1} + \sqrt{\sigma_{22} - (\sigma_{12})^2/\sigma_{11}}\varepsilon_{i2} > 0 \end{array} \right\} \tag{4}$$

From expression (4), different values for (β, Σ) give the same value for the probability. In particular, for two arbitrary positive constants (c, d), the value of the parameters $\{c(\beta_1, \delta_{11}), d(\beta_2, \delta_{21}, \delta_{22})\}$ give the same value for the probability as $\{\beta_1, \delta_{11}, \beta_2, \delta_{21}, \delta_{22}\}$. Hence, the model is not identified.

The most common normalization in the literature is to fix $\sigma_{11} = \sigma_{22} = 1$ (e.g. Chib and Greenberg 1998). However, the following section shows that it is more convenient from the point of view of computational tractability to choose $\sigma_{11} = \sigma_{22} - (\sigma_{12})^2 / \sigma_{11} = 1$. From expression (4), both normalizations make the model identified without imposing any unnecessary restrictions upon the parameters.

2.1.2 Sampling the Variance-Covariance Matrix.

In the Bayesian approach, model specification is completed by providing a prior distribution for the parameters. Let the prior for Σ be an Inverted Wishart $IW(2, df_0, K_0)$ distribution conditional to the restriction that $\sigma_{11} = \sigma_{22} - (\sigma_{12})^2 / \sigma_{11} = 1$. That is, given a matrix Σ that satisfies the restriction, the kernel of the prior is:

$$|\Sigma|^{-df_0/2} \exp\left(-1/2tr\left(\Sigma^{-1} K_0\right)\right)$$

The expected value of the unrestricted prior is $\frac{1}{df_0 - 6} K_0$. The definition of inverted Wishart distribution used here is the one described in Press (1986, pp. 117).

The conditional posterior of Σ given parameters β and latent data $\{y_{it}^* : t = 1, ..., T\}_{i=1}^T$ is an inverted Wishart $IW(2, df, K)$ with the restriction that $\sigma_{11} = \sigma_{22} - (\sigma_{12})^2 / \sigma_{11} = 1$. The parameters of this inverted Wishart are $df = df_0 + N$ and $K = K_0 + \sum_{i=1}^N e_i e_i^T$.

The following theorem, which can be found in Bauwens et al. (1999, pages 305-306), is useful to sample Σ, conditional on the normalization $\delta_{11} = \delta_{22} = 1$.

Theorem 1 *Let Σ be distributed as an $IW(d, df, G)$, and be partitioned as $\Sigma = (\Sigma_{ij})$, i,j=1,2, being Σ_{11} a $q \times q$ matrix. Define $\Sigma_{22 \cdot 1} = \Sigma_{22} - \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12}$, then*

$$\begin{aligned} &1) \quad \Sigma_{22 \cdot 1} | \Sigma_{11} \sim IW(d - q, df - q, G_{22 \cdot 1}) \\ &2) \quad \Sigma_{12} | (\Sigma_{22 \cdot 1}, \Sigma_{11}) \sim \quad MN\left(\Sigma_{11} (G_{11})^{-1} G_{12}, \Sigma_{11} (G_{11})^{-1} \Sigma_{22 \cdot 1} \Sigma_{11}\right) \end{aligned}$$

where MN refers to a matrix normal distribution.

Theorem 1 states that the conditional posterior of σ_{12} given β and all latent data y_{it}^* is a normal distribution and hence it can be sampled directly.

2.2 Dimension larger than 2.

The proposed normalization in the bivariate probit is to set the variance of e_{i1} (σ_{11}) and the conditional variance of e_{i2} given e_{i1} $\left(\sigma_{22} - (\sigma_{12})^2 / \sigma_{11}\right)$ both equal to one. Or in other words, to fix the elements in the diagonal of the cholesky decomposition equal to one. Similarly, in the case of $T \geq 2$ the proposed normalization is

$$Var(e_{i1}) = Var(e_{i2}|e_{i1}) = Var(e_{i3}|e_{i2}, e_{i1}) = ... = Var\left(e_{iT}|e_{i(T-1)}, ..., e_{i1}\right) = 1 \tag{5}$$

This normalization can also be shown to be the same as fixing the elements in the diagonal of the Cholesky decomposition equal to one. The following lemma gives an statistical interpretation to each of the elements of the Cholesky decomposition. The proof of lemma 1 is in appendix 4.

Lemma 1 *If e_i follows a $N(0, \Sigma)$, and Δ is the lower triangular cholesky decomposition, then*

$$\Delta = \begin{pmatrix} \sqrt{\sigma_{11}} & 0 & 0 & 0 & & 0 \\ \frac{\sigma_{12}}{\sqrt{\sigma_{11}}} & \sqrt{\sigma_{22 \cdot 1}} & 0 & 0 & & 0 \\ \frac{\sigma_{13}}{\sqrt{\sigma_{11}}} & \frac{\sigma_{23 \cdot 1}}{\sqrt{\sigma_{22 \cdot 1}}} & \sqrt{\sigma_{33 \cdot 12}} & 0 & & 0 \\ \frac{\sigma_{14}}{\sqrt{\sigma_{11}}} & \frac{\sigma_{24 \cdot 1}}{\sqrt{\sigma_{22 \cdot 1}}} & \frac{\sigma_{34 \cdot 12}}{\sqrt{\sigma_{33 \cdot 12}}} & \sqrt{\sigma_{44 \cdot 123}} & & 0 \\ & & & & \cdots & \\ \frac{\sigma_{1T}}{\sqrt{\sigma_{11}}} & \frac{\sigma_{2T \cdot 1}}{\sqrt{\sigma_{22 \cdot 1}}} & \frac{\sigma_{3T \cdot 12}}{\sqrt{\sigma_{33 \cdot 12}}} & \frac{\sigma_{4T \cdot 123}}{\sqrt{\sigma_{44 \cdot 123}}} & \cdots & \sqrt{\sigma_{TT \cdot 123...(T-1)}} \end{pmatrix}$$

where $\sigma_{tt \cdot vwz} = Var\left(e_{it}|e_{iv}, e_{iw}, e_{iz}\right)$ *and* $\sigma_{th \cdot vwz} = Cov\left(e_{it}, e_{ih}|e_{iv}, e_{iw}, e_{iz}\right)$.

Hence, restriction (5) is equivalent to fixing the diagonal elements of the Cholesky decomposition to one. In addition, from (6) this restriction identifies the model, not imposing any unnecessary restrictions upon the parameters.

Let Σ_{jj} be the sub-matrix of Σ containing the first j rows and the first j columns of Σ and K_{jj} the corresponding sub-matrix of K. Let Σ_j be the vertical vector containing the first $(j-1)$ rows in the jth column and let K_j be the corresponding sub-matrix of K. Following theorem 1, if Σ follows an unrestricted $IW\left(T, df, K\right)$, then

- Σ_T conditional on $\Sigma_{(T-1)(T-1)}$ and $\sigma_{TT \cdot 12...(T-1)}$ follows a normal distribution
- $\sigma_{TT \cdot 12...(T-1)}$ is independent of $\Sigma_{(T-1)(T-1)}$ and follows an inverted Wishart distribution.
- $\Sigma_{(T-1)(T-1)}$ follows an inverted Wishart distribution.

The third property holds because the marginal distribution of sub-matrices centered in the diagonal is also an inverted Wishart distribution (Press 1986, pp. 118-119). Since $\sigma_{TT \cdot 12...(T-1)}$ is independent of $\Sigma_{(T-1)(T-1)}$, conditioning on $\sigma_{TT \cdot 12...(T-1)} = 1$ does not change the marginal distribution of $\Sigma_{(T-1)(T-1)}$.

Consider now that Σ follows an $IW\left(T, df, K\right)$ with the restriction that $\sigma_{11} = \sigma_{22 \cdot 1} = ... = \sigma_{TT \cdot 12...(T-1)} = 1$. By applying the above argument recursively, the marginal distribution of Σ_2 given the restriction is a normal distribution. In addition, the distribution of Σ_n conditional on $\Sigma_{(n-1)(n-1)}$ and $\sigma_{nn \cdot 12...(n-1)}$ is also a normal distribution, for $2 \leq n \leq T$. Appendix 1 gives full detail of the distributions involved in this decomposition of the inverted Wishart density.

The following algorithm describes how Σ can be sampled from its conditional posterior distribution:

Algorithm 1 *Step 1: Sample* σ_{12} *conditional on* σ_{11} *and* $\sigma_{22 \cdot 1}$ *from a*

$$N\left(\sigma_{11} K_{11}^{-1} K_2, \left(\sigma_{11}\right)^2 K_{11}^{-1} \sigma_{22 \cdot 1}\right)$$

Step 2: Fix $\sigma_{22} = 1 + \left(\sigma_{12}\right)^2$
Step 3: Sample Σ_3 *conditional on* Σ_{22} *and* $\sigma_{33 \cdot 12}$ *from a*

$$N\left(\Sigma_{22} K_{22}^{-1} K_3, \sigma_{33 \cdot 12} \Sigma_{22} K_{22}^{-1} \Sigma_{22}\right)$$

Step 4: Fix $\sigma_{33} = 1 + \Sigma_3^T \Sigma_{22}^{-1} \Sigma_3$.
Step 2(n-1)-1: Sample Σ_n *conditional on* $\Sigma_{(n-1)(n-1)}$ *and* $\sigma_{nn \cdot 12...(n-1)}$ *from a*

$$N\left(\Sigma_{(n-1)(n-1)} K_{(n-1)(n-1)}^{-1} K_n, \sigma_{nn \cdot 12...(n-1)} \Sigma_{(n-1)(n-1)} K_{(n-1)(n-1)}^{-1} \Sigma_{(n-1)(n-1)}\right)$$

Step 2(n-1): Fix $\sigma_{nn} = 1 + \Sigma_n^T \Sigma_{(n-1)(n-1)}^{-1} \Sigma_n$.

2.3 An alternative normalization.

This section describes how to transform the estimated values for (β, Σ) if another normalization is chosen. In particular, instead of normalization (5), one might be interested in choosing the more widely used normalization:

$$\sigma_{11} = \sigma_{22} = ... = \sigma_{TT} = 1 \tag{6}$$

A sample from the posterior of (β, Σ) given normalization (7) can be obtained by simply transforming the values sampled using algorithm 1. Let C_1 be the diagonal matrix of dimension T with diagonal equal to $\left(1/\sqrt{\sigma_{11}}, 1/\sqrt{\sigma_{22 \cdot 1}}, 1/\sqrt{\sigma_{33 \cdot 12}}, ..., 1/\sqrt{\sigma_{TT \cdot 12...(T-1)}}\right)$ and let C_2 be the diagonal matrix of dimension T

with diagonal equal to $\left(1/\sqrt{\sigma_{11}}, 1/\sqrt{\sigma_{22}}, 1/\sqrt{\sigma_{33}}, ..., 1/\sqrt{\sigma_{TT}}\right)$. The parameters that are identified with normalization (5) are $\left(1/\sqrt{\sigma_{11}}\beta_1, ..., 1/\sqrt{\sigma_{TT\cdot 12...(T-1)}}\beta_T, C_1 \Sigma C_1\right)$. When normalization (7) is used, the identified parameters are $\left(1/\sqrt{\sigma_{11}}\beta_1, ..., 1/\sqrt{\sigma_{TT}}\beta_T, C_2 \Sigma C_2\right)$.

Let $\left(\beta^k, \Sigma^k\right)_1$ be the kth value in the chain when normalization (5) is used. And let $\left(\beta^k, \Sigma^k\right)_2$ be the kth value in a chain in which normalization (7) is chosen. To obtain a sample from the posterior when normalization (7) is used, transform $\left(\beta^k, \Sigma^k\right)_1$ in the following way:

- Construct C_2^k as the diagonal matrix with diagonal equal to

$$\left(1/\sqrt{\sigma_{11}^k}, 1/\sqrt{\sigma_{22}^k}, 1/\sqrt{\sigma_{33}^k}, ..., 1/\sqrt{\sigma_{TT}^k}\right)$$

- Fix $\left(\beta^k, \Sigma^k\right)_2 = \left(1/\sqrt{\sigma_{11}^k}\beta_1, ..., 1/\sqrt{\sigma_{TT}^k}\beta_T, , C_2^k \Sigma^k C_2^k\right)$

3 Parameter Expansion Data Augmentation (PX-DA).

Parameter Expansion Data Augmentation (Liu and Wu 1998) as applied to the probit model, can be seen as sampling from a reparameterization of the latent data. This is shown in section 3.1.2. Section 3.2 explains how the PX-DA algorithm can be applied to the multivariate probit model. Section 3.3 compares the PX-DA algorithm with the proposed reparameterization in a general model. It is shown that in some cases both approaches do not lead to the same algorithm and that PX-DA algorithm is not invariant to reparameterizations.

3.1 The Univariate Probit Model.

3.1.1 Adding a Non-Identified Parameter to the Model (Liu and Wu 1998).

Liu and Wu (1998) describe their method as the introduction of a parameter α that is not identified in the model. In the probit case, they suggest to introduce the parameter in the following way:

$$\begin{aligned} \alpha y_{i1}^* &= X_{i11}\alpha\beta_{11} + X_{i12}\alpha\beta_{12} + X_{i13}\alpha\beta_{13} + ... + X_{i1k_1}\alpha\beta_{1k_1} + \alpha e_{i1} \\ & e_{i1} \sim N(0,1) \end{aligned}$$

Either a proper or improper prior might be placed on α. In the case of a proper density $h(\alpha)$ the algorithm is:

- Sample $(\alpha, \alpha Y_1^*, ..., \alpha Y_N^*)$ conditional on (β). That is, sample α from $h(\alpha)$ and $(\alpha Y_1^*, ..., \alpha Y_N^*)$ from truncated $N\left(X_{i11}\alpha\beta_{11} + X_{i12}\alpha\beta_{12} + ... + X_{i1k_1}\alpha\beta_{1k_1}, \alpha^2\right)$.
- Sample $(\alpha, \beta_{11}, \beta_{12}, ..., \beta_{1k_1})$ conditional on $(\alpha Y_1^*, ..., \alpha Y_N^*)$.

For any prior density of α that is independent of the prior of the slope parameters, the algorithm will converge to the desired posterior distribution. Consider a proper prior that in the limit is an improper prior. Because the algorithm converges to the same distribution for any prior, the limit of the kernel of the algorithm, as the prior tends to an improper prior, would also have the posterior distribution as stationary distribution. The prior $\pi(\alpha) \propto 1/\alpha$ is recommended because no proper prior would yield faster convergence. Since this prior can be seen as the limiting case of a proper prior $h(\alpha)$, then the transition kernel of the previous algorithm converges to:

Algorithm 2 *Step 1) Sample* $(Y_1^*, ..., Y_N^*)$ *from truncated* $N(X_{i11}\beta_{11} + X_{i12}\beta_{12} + ... + X_{i1k_1}\beta_{1k_1}, 1)$.
Step 2) Sample $(\alpha, \beta_{11}, \beta_{12}, ..., \beta_{1k_1})$ *conditional on* $(Y_1^*, ..., Y_N^*)$ *using kernel*

$$\exp\left\{-\frac{1}{2}(\beta_1 - \beta_0)^T V_0^{-1}(\beta_1 - \beta_0)\right\} \exp\left\{-\frac{1}{2\alpha^2}\sum_{i=1}^{N}(Y_i^* - X_{1i}\alpha\beta_1)^2\right\}(\alpha)^{-(N+1)}$$

where it has been assumed that β_1 follows a priori a $N(\beta_0, V_0)$.

3.1.2 A Different Interpretation for PX-DA.

The above algorithm can also be derived by doing a re-parameterization of the model, without adding any additional parameter to the model. Let

$$\left(Y_1^*, \overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right) = \left(Y_1^*, \frac{Y_2^*}{Y_1^*}, \frac{Y_3^*}{Y_1^*}, ..., \frac{Y_N^*}{Y_1^*}\right)$$

The posterior of $\left(\beta_1, Y_1^*, \overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right)$ is

$$\overline{\pi_{DA}}\left(\beta_1, Y_1^*, \overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}|Y_1, ..., Y_N\right) = \pi_{DA}\left(\beta_1, Y_1^*, Y_1^*\overline{Y_2^*}, Y_1^*\overline{Y_3^*}, ..., Y_1^*\overline{Y_N^*}|Y_1, ..., Y_N\right)|Y_1^*|^{N-1}$$

Consider the following algorithm:

Algorithm 3 *Step 1) Sample* $\left(Y_1^*, \overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right)$ *conditional on* β_1.
Step 2) Sample (Y_1^*, β_1) *conditional on* $\left(\overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right)$ *from kernel,*

$$\exp\left\{-\frac{1}{2}(\beta_1 - \beta_0)^T V_0^{-1}(\beta_1 - \beta_0)\right\} \exp\left\{-\frac{1}{2}(Y_1^* - X_{11}\beta_1)^2 - \frac{1}{2}\sum_{i=2}^{N}\left(\overline{Y_i^*}Y_1^* - X_{i1}\beta_1\right)^2\right\}(Y_1^*)^{N-1}$$

Note that the stationary distribution of this chain is the posterior distribution $\overline{\pi_{DA}}$.

This algorithm can be seen to be equal to algorithm (2). Let Y_1^v be the value of Y_1^* obtained in the first step of the algorithm. In the second step, define $\overline{Y_1^*} = Y_1^v\left(\frac{1}{Y_1^*}\right)$. The following proposition gives the conditional density of $\left(\overline{Y_1^*}, \beta_1\right)$.

Proposition 1 *The conditional density of* $\left(\overline{Y_1^*}, \beta_1\right)$ *given* $\left(\overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right)$ *is proportional to,*

$$\exp\left\{-\frac{1}{2}(\beta_1 - \beta_0)^T V_0^{-1}(\beta_1 - \beta_0)\right\} \exp\left\{-\frac{1}{2\left(\overline{Y_1^*}\right)^2}\left[\left(Y_1^v - X_{1i}\overline{Y_1^*}\beta_1\right)^2 + \sum_{i=2}^{N}\left(Y_1^v\overline{Y_i^*} - X_{1i}\overline{Y_1^*}\beta_1\right)^2\right]\right\}\left(\overline{Y_1^*}\right)^{-(N+1)}$$

Proof. The posterior conditional density of (Y_1^*, β_1) is proportional to:

$$\exp\left\{-\frac{1}{2}(\beta_1 - \beta_0)^T V_0^{-1}(\beta_1 - \beta_0)\right\} \exp\left\{-\frac{1}{2}(Y_1^*)^2\left[\left(1 - X_{11}\beta_1\frac{1}{Y_1^*}\right)^2 + \sum_{i=2}^{N}\left(\overline{Y_i^*} - X_{1i}\beta_1\frac{1}{Y_1^*}\right)^2\right]\right\}(Y_1^*)^{N-1}$$

Since the Jacobian of the transformation from Y_1^* to $\overline{Y_1^*}$ is $\left|Y_1^v 1/\left(\overline{Y_1^*}\right)^2\right|$ the conditional density of $\left(\overline{Y_1^*}, \beta_1\right)$ is proportional to:

$$\exp\left\{-\frac{1}{2}(\beta_1 - \beta_0)^T V_0^{-1}(\beta_1 - \beta_0)\right\} \exp\left\{-\frac{1}{2}\left(\frac{Y_1^v}{\overline{Y_1^*}}\right)^2\left[\left(1 - X_{11}\beta_1\frac{\overline{Y_1^*}}{Y_1^v}\right)^2 + \sum_{i=2}^{N}\left(\overline{Y_i^*} - X_{1i}\beta_1\frac{\overline{Y_1^*}}{Y_1^v}\right)^2\right]\right\}\left(\overline{Y_1^*}\right)^{-(N+1)}$$

which is equal to the kernel above. ■

Hence, both algorithms are equivalent, since they have the same transition kernel. $\left(Y_1^*, \overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right)$ in step 1 of algorithm 3 can be sampled by sampling $(Y_1^*, ..., Y_N^*)$ and then doing the transformation. Hence, latent data is sampled equivalently in both algorithms. As proposition 1 shows, the slope parameters are also sampled from the same distribution. Section 3.3 shows that, under some conditions, the more general PX-DA algorithm can also be seen as sampling from a reparameterization.

The conditional distribution of $(\alpha, \beta_{11}, \beta_{12}, ..., \beta_{1k_1})$ given $(Y_1^*, ..., Y_N^*)$ is not of standard form, unless the prior distribution of β_1 is an improper flat prior $\pi(\beta_1) \propto 1$. Chen and Shao (1998) give general conditions for the existence of the posterior distribution in a probit model if the improper prior $\pi(\beta_1) \propto 1$ is used. As Liu and Wu (1998) note, when this prior is used, algorithm (2), and hence algorithm (3), reduces to:

- Sample $(Y_1^*, ..., Y_N^*)$ from truncated $N(X_{i11}\beta_{11} + X_{i12}\beta_{12} + ... + X_{i1k_1}\beta_{1k_1}, 1)$.
- Sample α^2 conditional on $(\alpha Y_1^*, ..., \alpha Y_N^*)$ from an inverted Wishart $IW(N, 1, K)$, with N degrees of freedom and
$$K = \sum_{i=1}^{N} \left[(Y_i^* - X_{i1}\mu)^2\right]$$
where $\mu = \left(\sum_{i=1}^N X_{i1}^T X_{i1}\right)^{-1} \sum_{i=1}^N X_{i1}^T Y_i^*$ is the OLS estimator when Y_i^* is observed.
- Divide all elements in $(Y_1^*, ..., Y_N^*)$ by α.
- Sample β_1 from a normal with mean μ and variance-covariance matrix $\left(\sum_{i=1}^N X_{i1}^T X_{i1}\right)^{-1}$.

Note that a standard Gibbs sampling algorithm consists in steps 1 and 4. Algorithms (2) and (3) add two steps, dividing all latent data by a random factor to make the algorithm moves faster. Algorithm (3) achieves that by conditioning on the ratio of Y_i^* to Y_1^*. Hence, if Y_1^* changes by a factor α then the rest of latent data has to vary by the same amount to keep the ratio constant.

3.2 PX-DA in the Multivariate Probit Model.

3.2.1 Improper Prior Case, Liu (2001)

Liu (2001) explains how the PX-DA algorithm can be applied to the Multivariate Probit Model using an improper prior for Σ. The suggestion is to introduce T non-identified parameters $(\alpha_1, ..., \alpha_T)$ into the model in the following way:

$$\alpha_t y_{it}^* = \alpha_t X_{it}\beta_t + \alpha_t e_{it} \qquad i = 1, ..., N \qquad t = 1, ..., T \tag{7}$$

The random vector $e_i = (e_{i1}, ..., e_{iT})^T$ follows a $N(0, \Sigma)$. In order to identify the model all elements in the diagonal of Σ are fixed to be one. It was assumed that the number of parameters in each equation is the same, $k_1 = k_2 =, ..., k_T = k$. A proper prior and an improper prior was specified for $\beta = \left(\beta_1^T, \beta_2^T, ..., \beta_T^T\right)^T$ and Σ, respectively.

$$\begin{aligned} \pi(\Sigma) &\propto |\Sigma|^{-\frac{(T+1)}{2}} \\ \pi(\beta|\Sigma) &\propto N_{TK}\left(0, \Omega^{-1} \otimes \Sigma\right) \end{aligned}$$

where Λ is a $T \times T$ known diagonal matrix. From the results in Liu and Wu (1998), the recommended prior for α is $\pi(\alpha) \propto \left(\prod_{t=1}^T \alpha_t\right)^{-(T+1)/2}$.

The grouping strategy followed was:

- Generate $(Y_1^*, ..., Y_N^*)$ conditional on (β, Σ).
- Generate β conditional on $(Y_1^*, ..., Y_N^*, \Sigma)$.

- Generate (α, Σ) conditional on $(Y_1^*, ..., Y_N^*, \beta)$.

By specifying the above improper prior on Σ, (α, Σ) can be sampled directly from their posterior distribution. It was argued that this algorithm not only avoids the need for a Metropolis Hastings algorithm, but also speeds up convergence.

However, it has not been proved in the literature whether the improper prior for Σ leads to a proper posterior and hence to a valid Bayesian analysis. Some improper priors result in an improper posterior in similar models. For instance, Natarajan and McCulloch (1995) show that in a probit model with random effects, a prior $1/\sigma_u$ for the variance of the individual effects implies that the posterior is improper. Hence, it is important to specify a proper prior when the implications of an improper prior are not known.

If a proper prior, such as a restricted inverted Wishart, is specified on the restricted Σ and the same improper prior is specified on α, the conditional distribution of (α, Σ) does not have a standard form, and a Metropolis step seems to be necessary. Next section and section 5 suggest possibilities for updating the latent data marginally on some parameters.

3.2.2 Proper Prior Case

Consider the following reparameterization of the model:

$$\left(y_{1t}^*, \overline{y_{2t}^*}, \overline{y_{3t}^*}, ..., \overline{y_{Nt}^*}\right) = \left(y_{1t}^*, \frac{y_{2t}^*}{y_{1t}^*}, \frac{y_{3t}^*}{y_{1t}^*}, ..., \frac{y_{Nt}^*}{y_{1t}^*}\right) \qquad \text{for } t = 1, ..., T$$

As a normalization, assume that the diagonal elements of the cholesky decomposition of Σ are restricted to be one. Assume that Σ and β are independent a priori, with a $N(\beta_0, V_0)$ and a restricted $IW(T, df_0, K_0)$ as prior densities. One possibility to speed up convergence is given by the following algorithm:

- **Algorithm 4** *Step1) Generate* $\left\{\left(y_{1t}^*, \overline{y_{2t}^*}, \overline{y_{3t}^*}, ..., \overline{y_{Nt}^*}\right) : t = 1, ..., T\right\}$ *conditional on* (β, Σ) *using truncated normal distributions.*

  Step 2) Sample β *conditional on* $\left\{\left(y_{1t}^*, \overline{y_{2t}^*}, \overline{y_{3t}^*}, ..., \overline{y_{Nt}^*}\right) : t = 1, ..., T\right\}$ *and* Σ *from a normal distribution.*

  Step 3) Generate $(y_{11}^*, y_{12}^*, ..., y_{1T}^*)$ *conditional on* β *and* $\left\{\left(\overline{y_{2t}^*}, \overline{y_{3t}^*}, ..., \overline{y_{Nt}^*}\right) : t = 1, ..., T\right\}$ *but not on* Σ *using a Metropolis algorithm.*

  Step 4) Sample Σ *conditional on* $\left\{\left(y_{1t}^*, \overline{y_{2t}^*}, \overline{y_{3t}^*}, ..., \overline{y_{Nt}^*}\right) : t = 1, ..., T\right\}, \beta$ *using algorithm (1).*

The reason why the algorithm is faster is because some characteristics of the latent data, $(y_{11}^*, y_{12}^*, ..., y_{1T}^*)$, are sampled marginally on Σ. Hence, the pernicious effect of large correlations between Σ and latent data is diminished. Another possibility, that is not explored in this paper, is to sample $(y_{11}^*, y_{12}^*, ..., y_{1T}^*, \beta)$ jointly conditioning on Σ. This alternative would be preferable if the correlation of the latent data with slope parameters is larger than the correlation with Σ.

Appendix (2) gives full detail of the distributions that are used in step 3 and 4. Step 3 can be repeated several times to increase the probability of accepting a new value. Note that in this algorithm new values of Σ are always generated, even if none of the candidate values for $(y_{11}^*, y_{12}^*, ..., y_{1T}^*)$ was accepted. This makes the algorithm move faster, since each iteration generates different values for Σ. Note that this is not a standard Metropolis step to sample $(y_{11}^* y_{12}^*, ..., y_{1T}^*, \Sigma)$, but the stationary distribution of this chain is still the joint posterior density of $\left(\left\{\left(y_{1t}^*, \overline{y_{2t}^*}, \overline{y_{3t}^*}, ..., \overline{y_{Nt}^*}\right) : t = 1, ..., T\right\}, \Sigma, \beta\right)$. It can also be checked that the chain is Harris recurrent, since all distributions used to generate the parameters are strictly positive and continuous over all the parameter space (Tierney 1994, Theorem 1). Hence, the algorithm converges monotonically to the posterior distribution.

3.3 PX-DA in a General Model.

The intuitive idea underlying the PX-DA algorithm is that the latent data is transformed at each iteration. The interpretation given in section 3.1.2 and 3.2.2 is that the magnitud of the latent data is sampled

conditioning on the relative size of the latent data, and marginally on some parameters. By not conditioning on some parameters that are highly correlated with the latent data, the convergence is faster.

In the probit model, the latent data is transformed by dividing it by a random factor. However, different transformations of the latent data might be more appropriate for other models. This section is concerned with general transformations of the latent data. But it is thought in terms of sampling a feature of the latent data conditioning on some characteristics of the latent data. And hence, it is interpreted as a reparameterization.

Suppose that latent data is used in some arbitrary model, and that we are interested in transforming the latent data according to the function $\widetilde{Y_i^*} = t_\alpha(Y_i^*)$. Y_i^* represents the latent data for observation i, and $\widetilde{Y_i^*}$ the latent data after the transformation. In the previous section $t_\alpha(Y_i^*) = Y_i^*/\alpha$, so that latent data was transformed with a linear transformation. Let θ contain the rest of parameters in the model.

Assume that t_α is a continuos function, with a differentiable inverse function. Also assume that for any two values a_1 and a_2 there is a unique value for α such that $a_1 = t_\alpha(a_2)$. Let $\alpha = h(a_1, a_2)$ be the function that yields the unique value of α that makes $a_1 = t_\alpha(a_2)$. That is, $h(a_1, a_2)$ is the function satisfying $a_1 = t_{h(a_1,a_2)}(a_2)$.

Consider the following parameterization:

$$\left(Y_1^*, \overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right) = \left(Y_1^*, h(Y_2^*, Y_1^*), h(Y_3^*, Y_1^*), ..., h(Y_N^*, Y_1^*)\right)$$

and the following algorithm:

- Generate $\left(Y_1^*, \overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right)$ conditional on θ.
- Generate Y_1^* conditioning on $\left(\overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right)$ but not on θ.
- Generate θ conditional on $\left(Y_1^*, \overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right)$.

The second step makes the algorithm faster, because it samples a characteristic of the latent data marginally on θ. If we also conditioned on θ the convergence speed would be the same, but with a higher computational cost. In addition, as the following proposition shows, the second step can be seen to be equal to transform all latent data with the function $t_\alpha(.)$.

Proposition 2 *Let* $(Y_1^v, Y_2^v, Y_3^v, ..., Y_N^v)$ *be the value of* $(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*)$ *obtained in the first step of the algorithm. And let* $\left(\widetilde{Y_1^*}, \widetilde{Y_2^*}, \widetilde{Y_3^*}, ..., \widetilde{Y_N^*}\right)$ *be the implied value of* $(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*)$ *after the second step. There exists a value* α_n *such that:*

$$\left(\widetilde{Y_1^*}, \widetilde{Y_2^*}, \widetilde{Y_3^*}, ..., \widetilde{Y_N^*}\right) = \left(t_{\alpha_n}(Y_1^v), t_{\alpha_n}(Y_2^v), t_{\alpha_n}(Y_3^v), ..., t_{\alpha_n}(Y_N^v)\right)$$

Proof. Let α_n be the value of α such that $\widetilde{Y_1^*} = t_{\alpha_n}(Y_1^v)$. Since we condition on $h(Y_2^*, Y_1^*)$, it must be that $h(Y_2^v, Y_1^v) = h\left(\widetilde{Y_2^*}, \widetilde{Y_1^*}\right)$. Let $\alpha_o = h(Y_2^v, Y_1^v)$. The following relations hold:

$$\begin{aligned} Y_2^v &= t_{\alpha_0}(Y_1^v) \\ \widetilde{Y_2^*} &= t_{\alpha_0}\left(\widetilde{Y_1^*}\right) = t_{\alpha_0}(t_{\alpha_n}(Y_1^v)) = t_{\alpha_n}(t_{\alpha_0}(Y_1^v)) = t_{\alpha_n}(Y_2^v) \end{aligned}$$

where the property that the composition of functions is commutative has been used (i.e. $t_{\alpha_n}(t_{\alpha_0}(Y_1^v)) = t_{\alpha_0}(t_{\alpha_n}(Y_1^v))$). Since $\widetilde{Y_2^*} = t_{\alpha_n}(Y_2^v)$, all latent data has been transformed with the function $t_{\alpha_n}(.)$, and $\alpha_n = h\left(\widetilde{Y_1^*}, Y_1^v\right)$ has been generated at random. ■

Let $\pi_{DA}(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*, \theta | Y_1, Y_2, Y_3, ..., Y_N)$ be the posterior density of $(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*, \theta)$. The posterior density in the transformed model is:

$$\pi_{DA}\left(Y_1^*, t_{\overline{Y_2^*}}(Y_1^*), t_{\overline{Y_3^*}}(Y_1^*), ..., t_{\overline{Y_N^*}}(Y_1^*), \theta | Y_1, Y_2, Y_3, ..., Y_N\right) \left| \prod_{i=2}^{N} \frac{\partial t_{\overline{Y_i^*}}(Y_1^*)}{\partial \overline{Y_i^*}} \right|$$

Let Y_1^v be the value for Y_1^* obtained in the first step of the algorithm, and $\overline{Y_1^*} = h\left(Y_1^*, Y_1^v\right)$. In order to compare the PX-DA algorithm with the reparameterization, the following proposition yields an expression for the conditional posterior density of $\left(\overline{Y_1^*}, \theta\right)$ given $\left(\overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right)$.

Proposition 3 *The conditional posterior density of* $\left(\overline{Y_1^*}, \theta\right)$ *given* $\left(\overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}\right)$ *is proportional to:*

$$\pi_{DA}\left(t_{\overline{Y_1^*}}\left(Y_1^v\right), t_{\overline{Y_1^*}}\left(Y_2^v\right), t_{\overline{Y_1^*}}\left(Y_3^v\right), ..., t_{\overline{Y_1^*}}\left(Y_N^v\right), \theta | Y_1, Y_2, Y_3, ..., Y_N\right) \left| \frac{\partial t_{\overline{Y_1^*}}\left(Y_1^v\right)}{\partial \overline{Y_1^*}} \prod_{i=2}^{N} \frac{\partial t_{\overline{Y_1^*}}\left(Y_i^v\right)}{\partial Y_i^v} \right|$$

Proof. The posterior density of $\left(\overline{Y_1^*}, \overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*}, \theta\right)$ is proportional to:

$$\pi_{DA}\left(t_{\overline{Y_1^*}}\left(Y_1^v\right), t_{\overline{Y_2^*}}\left(t_{\overline{Y_1^*}}\left(Y_1^v\right)\right), t_{\overline{Y_3^*}}\left(t_{\overline{Y_1^*}}\left(Y_1^v\right)\right), ..., t_{\overline{Y_N^*}}\left(t_{\overline{Y_1^*}}\left(Y_1^v\right)\right), \theta | Y_1, Y_2, Y_3, ..., Y_N\right) \left| \frac{\partial t_{\overline{Y_1^*}}\left(Y_1^v\right)}{\partial \overline{Y_1^*}} \prod_{i=2}^{N} \frac{\partial t_{\overline{Y_i^*}}\left(t_{\overline{Y_1^*}}\left(Y_1^v\right)\right)}{\partial \overline{Y_i^*}} \right|$$

Since the composition of functions is commutative, this last expression can be written as:

$$\pi_{DA}\left(t_{\overline{Y_1^*}}\left(Y_1^v\right), t_{\overline{Y_1^*}}\left(t_{\overline{Y_2^*}}\left(Y_1^v\right)\right), t_{\overline{Y_1^*}}\left(t_{\overline{Y_3^*}}\left(Y_1^v\right)\right), ..., t_{\overline{Y_1^*}}\left(t_{\overline{Y_N^*}}\left(Y_1^v\right)\right), \theta | Y_1, Y_2, Y_3, ..., Y_N\right) \left| \frac{\partial t_{\overline{Y_1^*}}\left(Y_1^v\right)}{\partial \overline{Y_1^*}} \prod_{i=2}^{N} \frac{\partial t_{\overline{Y_1^*}}\left(t_{\overline{Y_i^*}}\left(Y_1^v\right)\right)}{\partial \overline{Y_i^*}} \right|$$

If we condition on $\left(\overline{Y_2^*}, ..., \overline{Y_N^*}\right)$ so that these values remain constant, then $t_{\overline{Y_i^*}}\left(Y_1^v\right) = Y_i^v$, for $i = 2, ..., N$. Hence, the posterior density of $\left(\overline{Y_1^*}, \theta\right)$ given $\left(\overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*},\right)$ is proportional to:

$$\pi_{DA}\left(t_{\overline{Y_1^*}}\left(Y_1^v\right), t_{\overline{Y_1^*}}\left(Y_2^v\right), t_{\overline{Y_1^*}}\left(Y_3^v\right), ..., t_{\overline{Y_1^*}}\left(Y_N^v\right), \theta | Y_1, Y_2, Y_3, ..., Y_N\right) \left| \frac{\partial t_{\overline{Y_1^*}}\left(Y_1^v\right)}{\partial \overline{Y_1^*}} \prod_{i=2}^{N} \frac{\partial t_{\overline{Y_1^*}}\left(t_{\overline{Y_i^*}}\left(Y_1^v\right)\right)}{\partial \overline{Y_i^*}} \right|$$

Using the chain rule for derivatives of composed functions the following equality holds:

$$\frac{\partial t_{\overline{Y_1^*}}\left(t_{\overline{Y_i^*}}\left(Y_1^v\right)\right)}{\partial \overline{Y_i^*}} = \frac{\partial t_{\overline{Y_1^*}}\left(t_{\overline{Y_i^*}}\left(Y_1^v\right)\right)}{\partial t_{\overline{Y_i^*}}\left(Y_1^v\right)} \frac{\partial t_{\overline{Y_i^*}}\left(Y_1^v\right)}{\partial \overline{Y_i^*}} = \frac{\partial t_{\overline{Y_1^*}}\left(Y_i^v\right)}{\partial Y_i^v} \frac{\partial t_{\overline{Y_i^*}}\left(Y_1^v\right)}{\partial \overline{Y_i^*}}$$

Since $\partial t_{\overline{Y_i^*}}\left(Y_1^v\right) / \partial \overline{Y_i^*}$ does not depend on $\overline{Y_1^*}$, the posterior density of $\left(\overline{Y_1^*}, \theta\right)$ given $\left(\overline{Y_2^*}, \overline{Y_3^*}, ..., \overline{Y_N^*},\right)$ is proportional to:

$$\pi_{DA}\left(t_{\overline{Y_1^*}}\left(Y_1^v\right), t_{\overline{Y_1^*}}\left(Y_2^v\right), t_{\overline{Y_1^*}}\left(Y_3^v\right), ..., t_{\overline{Y_1^*}}\left(Y_N^v\right), \theta | Y_1, Y_2, Y_3, ..., Y_N\right) \left| \frac{\partial t_{\overline{Y_1^*}}\left(Y_1^v\right)}{\partial \overline{Y_1^*}} \prod_{i=2}^{N} \frac{\partial t_{\overline{Y_1^*}}\left(Y_i^v\right)}{\partial Y_i^v} \right|$$

■

The PX-DA algorithm (Liu and Wu 1999, page 1269) is:

- Sample $\left(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*\right)$ conditional on θ from its conditional posterior density, which is proportional to $\pi_{DA}\left(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*, \theta | Y_1, Y_2, Y_3, ..., Y_N\right)$.
- Sample (α, θ) given $\left(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*\right) = \left(Y_1^v, Y_2^v, Y_3^v, ..., Y_N^v\right)$, from its conditional density:

$$\pi_{DA}\left(t_\alpha\left(Y_1^v\right), t_\alpha\left(Y_2^v\right), t_\alpha\left(Y_3^v\right), ..., t_\alpha\left(Y_N^v\right), \theta | Y_1, Y_2, Y_3, ..., Y_N\right) H(\alpha) \left| \prod_{i=1}^{N} \frac{\partial t_\alpha\left(Y_i^v\right)}{\partial Y_i^v} \right| \tag{8}$$

where $H(\alpha)$ is the Haar prior for α. The Haar prior is a function such as for any value (a, b, d)

$$\int_a^b H(\alpha)\, d\alpha = \int_{t_d(a)}^{t_d(b)} H(\alpha)\, d\alpha$$

Hence, both approaches lead to the same algorithm if and only if the Haar prior $H(\alpha)$ is proportional to:

$$\left| \frac{\partial t_\alpha (Y_1^v)}{\partial \alpha} \left(\frac{\partial t_\alpha (Y_1^v)}{\partial Y_1^v} \right)^{-1} \right| \tag{9}$$

Liu and Wu (1999) show that the PX-DA algorithm can be written as:

- Sample $(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*)$ conditional on θ from its conditional posterior density, which is proportional to $\pi_{DA}(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*, \theta | Y_1, Y_2, Y_3, ..., Y_N)$.
- Sample (α) from the marginal distribution of α implied by kernel (9).
- Transform all latent data with the function $t_\alpha(.)$. That is, fix $(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*) = (t_\alpha(Y_1^v), t_\alpha(Y_2^v), t_\alpha(Y_3^v), ..., t_\alpha(Y_N^v))$.
- Sample θ from its conditional density given $(Y_1^*, Y_2^*, Y_3^*, ..., Y_N^*)$.

From proposition 1, the algorithm that results from the re-parameterization also transforms the latent data with the function $t_{\overline{Y_1^*}}(Y_N^v)$, where $\overline{Y_1^*} = h(Y_1^*, Y_1^v)$. Hence, both algorithms are the same if both α and $\overline{Y_1^*}$ are sampled from the same density.

If $t_\alpha(Y_1^v) = Y_1^v/\alpha$, as in the previous section, the Haar prior is $H(\alpha) = 1/\alpha$, and both approaches lead to the same algorithm. If $t_\alpha(Y_1^v) = \alpha Y_1^v$, the Haar prior continues to be the same, and also in this case both interpretations are equivalent. Hence, for linear transformations of the latent data, $t_\alpha(Y_1^v) = \alpha Y_1^v + a$, where a is a known constant, both interpretations yield the same algorithm. However, for some transformations both methods do not coincide. The following example illustrates that the PX-DA algorithm is not invariant to reparameterizations.

Example 2 *Rewrite the univariate probit model as:*

$$q_{i1}^* = \exp(y_{i1}^*) = \exp(X_{i11}\beta_{11}) \exp(X_{i12}\beta_{12}) \exp(X_{i13}\beta_{13}) ... \exp(X_{i1k_1}\beta_{1k_1}) \exp(e_{i1})$$

$$\begin{aligned} y_{i1} &= 1 \qquad \textit{iff } q_{i1}^* > 1 \\ y_{i1} &= 0 \qquad \textit{iff } q_{i1}^* \leq 1 \end{aligned}$$

Following the motivation of Parameter Expansion Data Augmentation, let α be a non-identified parameter that enters in the model in the following way:

$$(q_{i1}^*)^\alpha = \exp(X_{i11}\beta_{11})^\alpha \exp(X_{i12}\beta_{12})^\alpha \exp(X_{i13}\beta_{13})^\alpha ... \exp(X_{i1k_1}\beta_{1k_1})^\alpha \exp(e_{i1})^\alpha$$

By taking logarithms, this model can be seen to be equal to model (8). Using the PX-DA algorithm, (β_1, α) are drawn from expression (9), with $t_\alpha(q_{i1}^*) = (q_{i1}^*)^{1/\alpha}$. By taking logarithms at both sides of $\widetilde{q_{i1}^*} = (q_{i1}^*)^{1/\alpha}$, this transformation can be seen to be equal to $\widetilde{y_i^*} = 1/\alpha (y_i^*)$. Hence, this transformation is equivalent to dividing the original latent data y_i^* by α. For both algorithms to give the same result, the haar prior $H(\alpha)$ should be proportional to:

$$\left| \frac{1}{\alpha^2} (q_1^v)^{1/\alpha} \ln(q_1^v) \left(\frac{1}{\alpha} (q_1^v)^{1/\alpha - 1} \right)^{-1} \right| \propto \frac{1}{\alpha}$$

where q_1^v is the value of q_1^* obtained in the first step of the algorithm.

However, it can be verified that $1/\alpha$ is not the Haar prior since for a general value of (a, b, d) the value of the integral

$$\int_a^b \frac{1}{\alpha} d\alpha$$

is not proportional to the integral:

$$\int_{a^d}^{b^d} \frac{1}{\alpha} d\alpha$$

The reparameterization that corresponds to this transformation of the latent data is:

$$\left(\overline{q_1^*}, \overline{q_2^*}, ..., \overline{q_N^*}\right) = \left(\frac{\ln\left(q_1^v\right)}{\ln\left(q_1^*\right)}, \frac{\ln\left(q_1^*\right)}{\ln\left(q_2^*\right)}, ..., \frac{\ln\left(q_1^*\right)}{\ln\left(q_N^*\right)}\right) = \left(\frac{y_1^v}{y_1^*}, \frac{y_1^*}{y_2^*}, ..., \frac{y_1^*}{y_N^*}\right) = \left(\overline{y_1^*}, \overline{y_2^*}, ..., \overline{y_N^*}\right)$$

Hence, if the data is transformed using a reparameterization, the algorithm is still the same as when the transformation $\tilde{y_i^*} = 1/\alpha\left(y_i^*\right)$ *is considered. However, the PX-DA algorithm changes when the parameterization changes.*

The previous example shows a non-linear transformation of the latent data for which both algorithms are not the same. However, this transformation can be seen as a linear transformation on the logs of the latent data. In contrast with the reparameterization, the PX-DA algorithm to transform the latent data non-linearly is not equivalent to the PX-DA algorithm to transform the log of latent data linearly, even though both transformations are the same in the example above.

4 Data Augmentation in the Multivariate Probit Model.

For simplicity in the exposition, we first focus in the case of the probit model, that is $T = 1$. Section 4.2 considers the multivariate case.

4.1 The Univariate Probit.

The intuitive idea for the algorithm is similar to the one in the previous section. But instead of multiplying the latent data by a random factor, the slope parameters will be multiplied marginally on the latent data. By not conditioning on the latent data, the algorithm is able to make larger moves.

Let $\beta_1 = (\beta_{11}, ..., \beta_{1k_1})$, and consider the following reparameterization of the model,

$$\begin{aligned} \overline{\beta_1} &= \left(\beta_{11}, \frac{\beta_{12}}{\beta_{11}}, \frac{\beta_{13}}{\beta_{11}}, ..., \frac{\beta_{1k_1}}{\beta_{11}}\right) = \left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right) \\ y_{i1}^* &= X_{i11}\beta_{11} + X_{i12}\overline{\beta_{12}}\beta_{11} + X_{i13}\overline{\beta_{13}}\beta_{11} + ... + X_{i1k_1}\overline{\beta_{1k_1}}\beta_{11} + e_{i1} \end{aligned}$$

The posterior distribution of $\overline{\beta_1}$ is given by the theorem of the change of variables,

$$\overline{\pi_M}\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}} | Y_1, ..., Y_N\right) = \pi_M\left(\beta_{11}, \overline{\beta_{12}}\beta_{11}, ..., \overline{\beta_{1k_1}}\beta_{11} | Y_1, ..., Y_N\right) \left|\beta_{11}\right|^{k_1 - 1}$$

where $\left|\beta_{11}^{k_1-1}\right|$ is the Jacobian of the transformation.

The proposed algorithm is,

- Sample $\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$ conditional on $(Y_1^*, ..., Y_N^*)$
- Sample β_{11} conditional on $\left(\overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$
- Sample $(Y_1^*, ..., Y_N^*)$ conditional on $\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$

Note that the stationary distribution of the chain is the posterior density of parameters and latent data. The conditional distribution of $(Y_1^*, ..., Y_N^*)$ is the same as in the algorithm proposed by Chib and Greenberg (1998). $\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$ can be sampled by generating $(\beta_{11}, \beta_{12}, ..., \beta_{1k_1})$ conditional on $(Y_1^*, ..., Y_N^*)$ and then transforming the variables $\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right) = (\beta_{11}, \beta_{12}/\beta_{11}, ..., \beta_{1k_1}/\beta_{11})$.

The conditional distribution of β_{11} given $\left(\overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$ does not have a standard form. Hence, a metropolis step can be used to generate β_{11}. The proposal density for the Metropolis step could be a

normal with mean and variance equal to the Maximum Likelihood estimation of β_{11}. Alternatively, β_{11} can be generated as in a random walk. Assume that the prior distribution of $(\beta_{11}, \beta_{12}, ..., \beta_{1k_1})$ is a $N(\beta_0, V_0)$. Let β_{11}^n denote the nth value of β_{11} in the chain. If a random walk is used, the algorithm to generate the (n+1)th value of $\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$ is:

Algorithm 5 *Step 1. Sample* $(\beta_{11}, \beta_{12}, ..., \beta_{1k_1})$ *from a* $N(\mu_p, V_p)$, *where* $\mu_p = V_p\left(\sum_{i=1}^N X_{i1}^T Y_i^* + V_0^{-1}\beta_0\right)$, *and* $V_p = \left(\sum_{i=1}^N X_{i1}^T X_{i1} + V_0^{-1}\right)^{-1}$.

Step 2. Fix $\left(\overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right) = (\beta_{12}/\beta_{11}, ..., \beta_{1k_1}/\beta_{11})$. *Generate a random scalar* v *from a distribution with density function* $f(v)$. *Fix* $\beta_{11}^{n+1} = v\beta_{11}^n$ *with probability*

$$\gamma = \min\left\{\frac{L\left(Y|v\beta_{11}^n, v\beta_{12}, ..., v\beta_{1k_1}\right)\pi\left(v\beta_{11}^n, v\beta_{12}, ..., v\beta_{1k_1}\right)}{L\left(Y|\beta_{11}^n, \beta_{12}, ..., \beta_{1k_1}\right)\pi\left(\beta_{11}^n, \beta_{12}, ..., \beta_{1k_1}\right)}\frac{f(1/v)}{f(v)}\left|v^{k_1}\right|, 1\right\}$$

and fix $\beta_{11}^{n+1} = \beta_{11}^n$ *with probability* $(1-\gamma)$, *where* $L\left(Y|\beta_{11}, \beta_{12}, ..., \beta_{1k_1}\right)$ *is the likelihood function:*

$$L(Y|\beta_1) = \prod_{i=1}^N \left(\Phi\left(-X_{1i}\beta_1\right)^{1-Y_i}\left(1-\Phi\left(-X_{1i}\beta_1\right)\right)^{Y_i}\right)$$

and $\pi\left(\beta_{11}, \beta_{12}, ..., \beta_{1k_1}\right)$ *is the prior density.*

Step 3. Sample Y_i^* *from a truncated* $N\left(X_{i11}\beta_{11} + X_{i12}\overline{\beta_{12}}\beta_{11} + ... + X_{i1k_1}\overline{\beta_{1k_1}}\beta_{11}, 1\right)$ *for all* $i = 1, ..., N$.

Step 2 of the algorithm can be repeated a number of times to increase the likelihood of acceptance of a new value. Note that algorithm (5) is the same as a Gibbs algorithm, with an additional step 2. Hence, it is just a Gibbs sampling algorithm, in which at each iteration all slope parameters might be multiplied by a random factor. Whenever a new candidate is accepted in step (2), all parameters move in the same direction. Step (2) accelerates the algorithm because it proposes a change of all parameters that is unconditional on the latent data. Conditioning on latent data makes the parameter move substantially slower.

The function $f(v)$ might be centered at 1, so that new candidates are drawn from a distribution centered at the old value. In addition, it might be desirable to restrict $f(v)$ to positive values, hence forcing new candidates to have the same sign as the previous value. An inverted gamma would play this role and it would be equivalent to drawing β_{11} conditioning not only on $\left(\overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$ but also on the sign of (β_{11}). In any case, the posterior density continues to be the stationary distribution of the chain, and it can be verified that the chain is Harris recurrent, using the conditions of Theorem 1 in Tierney (1994).

If $\phi_N\left(x; \widehat{\beta_{11}}, \widehat{sd1}\right)$ is the density function of a $N\left(\widehat{\beta_{11}}, \widehat{sd1}\right)$, then step (2) can be implemented as:

- Step 2'. Fix $\left(\overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right) = (\beta_{12}/\beta_{11}, ..., \beta_{1k_1}/\beta_{11})$. Generate a random scalar v from a distribution with density function $\phi_N\left(v; \widehat{\beta_{11}}, \widehat{sd1}\right)$. Fix $\beta_{11}^{n+1} = v$ with probability

$$\gamma' = \min\left\{\frac{L\left(Y|v, v\overline{\beta_{12}}, ..., v\overline{\beta_{1k_1}}\right)\pi\left(v, v\overline{\beta_{12}}, ..., v\overline{\beta_{1k_1}}\right)}{L\left(Y|\beta_{11}^n, \beta_{12}, ..., \beta_{1k_1}\right)\pi\left(\beta_{11}^n, \beta_{12}, ..., \beta_{1k_1}\right)}\frac{\phi_N\left(\beta_{11}^n; \widehat{\beta_{11}}, \widehat{sd1}\right)}{\phi_N\left(v; \widehat{\beta_{11}}, \widehat{sd1}\right)}\left|\left(\frac{v}{\beta_{11}^n}\right)^{k_1-1}\right|, 1\right\}$$

  and fix $\beta_{11}^{n+1} = \beta_{11}^n$ with probability $(1-\gamma')$.

4.2 The Multivariate Case.

The intuitive idea underlying the generalization to the multivariate probit model continues to be the same. At each iteration of the Gibbs sampler, the slope parameters of each equation are updated by a random factor. However, two distinctive features of the multivariate case should be noted. Firstly, the slope parameters in one equation have usually low correlation with the slope parameters in the rest of equations. Hence, it

does not seem a good idea to multiply the slope parameters in two different equations by the same factor. Such a movement would work best when parameters are highly correlated. Secondly, the likelihood of the multivariate probit cannot be easily calculated for $T > 3$. Hence, it is necessary to condition on some latent data to be able to carry out a metropolis step.

Consider the following re-parameterization of the model:

$$\begin{aligned}
\overline{\beta_1} &= \left(\beta_{11}, \frac{\beta_{12}}{\beta_{11}}, \frac{\beta_{13}}{\beta_{11}}, ..., \frac{\beta_{1k_1}}{\beta_{11}}\right) = \left(\beta_{11}, \overline{\beta_{12}}, \overline{\beta_{13}}, ..., \overline{\beta_{1k_1}}\right) \\
\overline{\beta_2} &= \left(\beta_{21}, \frac{\beta_{22}}{\beta_{21}}, \frac{\beta_{23}}{\beta_{21}}, ..., \frac{\beta_{2k_2}}{\beta_{21}}\right) = \left(\beta_{21}, \overline{\beta_{22}}, \overline{\beta_{23}}..., \overline{\beta_{2k_2}}\right) \\
&... \\
\overline{\beta_T} &= \left(\beta_{T1}, \frac{\beta_{T2}}{\beta_{T1}}, \frac{\beta_{T3}}{\beta_{T1}}, ..., \frac{\beta_{Tk_T}}{\beta_{T1}}\right) = \left(\beta_{T1}, \overline{\beta_{T2}}, \overline{\beta_{T3}}..., \overline{\beta_{Tk_T}}\right) \\
y^*_{i1} &= X_{i11}\beta_{11} + X_{i12}\overline{\beta_{12}}\beta_{11} + X_{i13}\overline{\beta_{13}}\beta_{11} + ... + X_{i1k_1}\overline{\beta_{1k_1}}\beta_{11} + e_{i1} \\
y^*_{i2} &= X_{i21}\beta_{21} + X_{i22}\overline{\beta_{22}}\beta_{21} + X_{i23}\overline{\beta_{23}}\beta_{21} + ... + X_{i2k_2}\overline{\beta_{2k_2}}\beta_{21} + e_{i2} \\
&... \\
y^*_{iT} &= X_{iT1}\beta_{T1} + X_{iT2}\overline{\beta_{T2}}\beta_{T1} + X_{i23}\overline{\beta_{T3}}\beta_{T1} + ... + X_{iTk_T}\overline{\beta_{Tk_T}}\beta_{T1} + e_{iT}
\end{aligned}$$

The grouping strategy in the algorithm is as follows:

- Sample $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}\right)$ conditional on $(Y^*_1, ..., Y^*_N, \Sigma)$.
- Sample Σ conditional on $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, Y^*_1, ..., Y^*_N\right)$
- For $t = 1, ..., T$ do:
  - Generate β_{t1} conditional on $\{y^*_{ik} : k \neq t\}^{i=1,...,N}$,$\{\beta_{tk} : k \neq 1\}$, $\left\{\overline{\beta_j} : j \neq t\right\}$, Σ.
  - Sample $\{y^*_{it}\}^{i=1,...,N}$ conditional on $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \Sigma\right)$ and $\{y^*_{ik} : k \neq t\}^{i=1,...,N}$

It can be checked that if the initial value of $(Y^*_1, ..., Y^*_N, \Sigma)$ is drawn from the marginal posterior distribution, then the value of $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, Y^*_1, ..., Y^*_N, \Sigma\right)$ after one iteration is also drawn from the posterior distribution. That is, the stationary distribution is the posterior density.

In the first part of the third step, β_{t1} is generated using a Metropolis step. This metropolis step can be repeated a number of times to increase the probability of accepting a new value. Thus, at each iteration of the gibbs algorithm, the slope parameters of each equation are updated not conditioning on the latent data of that equation. The first step and the second part of step 3 can be carried out as described in Chib and Greenberg (1998). The second step can be done following the algorithm described in section 2.

For some datasets, it might be worthwhile to save in computation time, at the cost of convergence speed. If this is the case, it might be better to update only the slope parameters of one equation in the first part of step 3. The equation whose parameters are updated can be chosen at random. That is, the third step can be substituted for the following scheme:

- Sample t using a uniform distribution defined in the set $\{1, 2, ..., T\}$.
- Generate β_{t1} conditional on $\{y^*_{ik} : k \neq t\}^{i=1,...,N}$,$\{\beta_{tk} : k \neq 1\}$, $\left\{\overline{\beta_j} : j \neq t\right\}$, Σ.
- Sample $(Y^*_1, ..., Y^*_N)$ conditional on $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \Sigma\right)$

This modification of the algorithm does not alter the stationary distribution, since it is a mixture of T kernels, each of them having the desired stationary distribution. As before, the Metropolis step can be repeated a number of times to increase the likelihood of acceptance.

If all parameters from all equations are updated in step 3, the algorithm is:

Algorithm 6 *Step 1. Sample* $(\beta_1, \beta_2, ..., \beta_T)$ *from a* $N(\mu_m, V_m)$, *where* $\mu_m = V_m\left(\sum_{i=1}^N X_i^T Y_i^* + V_0^{-1}\beta_0\right)$, *and* $V_m = \left(\sum_{i=1}^N X_i^T X_i + V_0^{-1}\right)^{-1}$.

Step 2. Sample Σ *using algorithm (1).*

Step 3. For $t = 1...T$ *do:*

- *Fix* $\left(\overline{\beta_{t2}}, ..., \overline{\beta_{tk_t}}\right) = (\beta_{t2}/\beta_{t1}, ..., \beta_{tk_t}/\beta_{t1})$. *Generate a random scalar* v *from a distribution with density function* $f_t(v)$. *Fix* $\beta_{t1}^{n+1} = v\beta_{t1}^n$ *with probability*

$$\gamma = \min\left\{\frac{L_t\left(Y|v\beta_{t1}^n, v\beta_{t2}, ..., v\beta_{tk_1}, \{y_{ik}^* : k \neq t\}^{i=1,...,N}, \left\{\overline{\beta_j} : j \neq t\right\}, \Sigma\right)\pi\left(v\beta_{t1}^n, v\beta_{t2}, ..., v\beta_{tk_1}\right)}{L_t\left(Y|\beta_{t1}^n, \beta_{t2}, ..., \beta_{tk_1}\{y_{ik}^* : k \neq t\}^{i=1,...,N}, \left\{\overline{\beta_j} : j \neq t\right\}, \Sigma\right)\pi\left(\beta_{t1}^n, \beta_{t2}, ..., \beta_{tk_1}\right)}\frac{f_t(1/v)}{f_t(v)}\left|v^{k_1}\right|, 1\right\}$$

and fix $\beta_{11}^{n+1} = \beta_{11}^n$ *with probability* $(1-\gamma)$.

- *Sample* $\{y_{it}^*\}^{i=1,...,N}$ *conditional on* $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \Sigma\right)$ *and* $\{y_{ik}^* : k \neq t\}^{i=1,...,N}$ *from truncated normals.*

The function $L_t(Y|.)$ is the likelihood that results when the latent data from all equations except for equation t are observed. New candidates for β_{t1} can also be generated with a proposal density centered on an approximation of the mode of the posterior distribution of β_{t1}. Let $\widehat{\beta_{t1}}, \widehat{sdt1}$ denote the maximum likelihood estimate and standard deviation of β_{t1} obtained by running a simple univariate probit model for equation t. Step 3 can be substituted by:

Step 3'. For $t = 1...T$ do:

- Fix $\left(\overline{\beta_{t2}}, ..., \overline{\beta_{tk_t}}\right) = (\beta_{t2}/\beta_{t1}, ..., \beta_{tk_t}/\beta_{t1})$. Generate a random scalar v from a $N\left(\sqrt{\sigma_{tt}}\widehat{\beta_{t1}}, \sqrt{\sigma_{tt}}\widehat{sd1t}\right)$, with density function $\phi_N\left(v; \sqrt{\sigma_{tt}}\widehat{\beta_{t1}}, \sqrt{\sigma_{tt}}\widehat{sd1t}\right)$. Let $\beta_{t1}^{n+1} = v$ with probability.

$$\gamma = \min\left\{\begin{array}{c}\frac{L_t\left(Y|v, v\overline{\beta_{12}}, ..., v\overline{\beta_{1k_1}}, \{y_{it}^* : t \neq k\}^{i=1,...,N}, \left\{\overline{\beta_j} : j \neq t\right\}, \Sigma\right)\pi\left(v, v\overline{\beta_{12}}, ..., v\overline{\beta_{1k_1}}\right)}{L_t\left(Y|\left(\beta_{t1}^n, \beta_{t1}^n\overline{\beta_{12}}, ..., \beta_{t1}^n\overline{\beta_{1k_1}}\right), \left\{y_{it}^* : t \neq k\right\}^{i=1,...,N}, \left\{\overline{\beta_j} : j \neq t\right\}, \Sigma\right)\pi\left(\beta_{t1}^n, \beta_{t1}^n\overline{\beta_{12}}, ..., \beta_{t1}^n\overline{\beta_{1k_1}}\right)} \\ \times\frac{\phi_N\left(\beta_{1t}^n; \sqrt{\sigma_{tt}}\widehat{\beta_{t1}}, \sqrt{\sigma_{tt}}\widehat{sd1t}\right)}{\phi_N\left(v; \sqrt{\sigma_{tt}}\widehat{\beta_{t1}}, \sqrt{\sigma_{tt}}\widehat{sd1t}\right)}\left|\left(\frac{v}{\beta_{t1}^n}\right)^{k_t-1}\right|, 1\end{array}\right\}$$

and fix $\beta_{t1}^{n+1} = \beta_{t1}^n$ with probability $(1-\gamma)$.

- Sample $\{y_{it}^*\}^{i=1,...,N}$ conditional on $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \Sigma\right)$ and $\{y_{ik}^* : k \neq t\}^{i=1,...,N}$ from truncated normals.

5 Updating Latent Data and Slope Parameters Jointly.

As noted above, data augmentation significantly increases the number of parameters in a model. However, this would not be a problem if sampling from the posterior joint distribution of parameters and latent data was possible. This is most often not possible, and instead, latent data are generated conditionally on the parameters and viceversa. High autocorrelations in the chain appear because, parameters which are highly correlated, such as latent data and slope parameters, are sampled as if they were independent.

In the context of the univariate probit, the marginal distribution of $(Y_1^*, ..., Y_N^*)$ unconditional on β_1 is multivariate normal of dimension N, with each component restricted to be positive or negative depending on the observed outcome Y_i. Philippe et al (2001) propose an algorithm to sample from such a distribution when N is small. Unfortunately, for large values of N the algorithm is computationally infeasible.

Updating all latent data and parameters at the same time, by multiplying them by a random factor, reduces the problem of sampling in different groups variables that are highly correlated. Consider the following re-parameterization of the multivariate probit model:

$$
\begin{aligned}
\overline{\beta_t} &= \left(\frac{\beta_{t1}}{y^*_{1t}}, \frac{\beta_{t2}}{y^*_{1t}}, \frac{\beta_{t3}}{y^*_{1t}}, ..., \frac{\beta_{tk_1}}{y^*_{1t}}\right) = \left(\overline{\beta_{t1}}, \overline{\beta_{t2}}, ..., \overline{\beta_{tk_1}}\right) \qquad t = 1, ..., T \\
\overline{Y^*_1} &= (y^*_{11}, y^*_{12}, ..., y^*_{1T}) \\
\overline{Y^*_i} &= \left(\frac{y^*_{i1}}{y^*_{11}}, \frac{y^*_{i2}}{y^*_{12}}, ..., \frac{y^*_{iT}}{y^*_{1T}}\right) = \left(\overline{y^*_{i1}}, \overline{y^*_{i2}}, ..., \overline{y^*_{iT}}\right) \qquad i = 2, ..., N \\
y^*_{1t}\overline{y^*_{it}} &= X_{it1}y^*_{1t}\overline{\beta_{t1}} + X_{it2}y^*_{1t}\overline{\beta_{t2}} + X_{it3}y^*_{1t}\overline{\beta_{t3}} + ... + X_{itk_1}y^*_{1t}\overline{\beta_{tk_t}} + e_{i1} \quad i = 2, ..., N \quad t = 1, ..., T \\
y^*_{1t} &= X_{it1}\overline{\beta_{t1}}y^*_{1t} + X_{it2}\overline{\beta_{t2}}y^*_{1t} + X_{it3}\overline{\beta_{t3}}y^*_{1t} + ... + X_{itk_T}\overline{\beta_{tk_T}}y^*_{1t} + e_{iT} \qquad t = 1, ..., T
\end{aligned}
$$

A possible algorithm to sample from this re-parameterization is:

Algorithm 7 *Step 1) Sample* $\left(\overline{Y^*_1}, ..., \overline{Y^*_N}\right)$ *conditional on* $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \Sigma\right)$ *from truncated normals.*

Step 2) Sample $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}\right)$ *conditional on* $\left(\overline{Y^*_1}, ..., \overline{Y^*_N}, \Sigma\right)$ *from a normal distribution.*

Step 3) Generate $(y^*_{11}, y^*_{12}, ..., y^*_{1T})$ *conditional on* $\left\{\left(\overline{y^*_{2t}}, \overline{y^*_{3t}}, ..., \overline{y^*_{Nt}}\right) : t = 1, ..., T\right\}$, $\left\{\overline{\beta_t}\right\}^{t=1,...,T}$ *with a Metropolis step.*

Step 4) Sample Σ *conditional on* $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \overline{Y^*_1}, ..., \overline{Y^*_N}\right)$ *using algorithm (1).*

The last step of the algorithm can be repeated several times to increase the likelihood of acceptance. The proposal density to generate new candidates for , $(y^*_{11}, y^*_{12}, ..., y^*_{1T})$ could be either a random walk or a density centered in maximum likelihood estimation of univariate probit models. Appendix 3 gives the density of $(y^*_{11}, y^*_{12}, ..., y^*_{1T})$ in step 3.

When $T = 1$, $(y^*_{11})^2$ conditional on $\left\{\left(\overline{y^*_{21}}, \overline{y^*_{31}}, ..., \overline{y^*_{N1}}\right)\right\}$, $\overline{\beta_1}$ follows a gamma distribution and can be sampled directly. This is a consequence of proposition 7, in appendix 3.

6 Sampling Marginally on the Latent Data: A Metropolis Step.

6.1 The Univariate Case.

As argued above, conditioning on the latent data significantly increases the number of variables and therefore slows down the algorithm. If the number of parameters in β_1 is small, a Metropolis step could be used. The Metropolis step works as follows. Let $r\left(\beta_1|\beta_1^n\right)$ be the proposal density, that is, a density that generates candidates for β_1 conditional on the previous value for β_1. These candidates will be accepted or rejected as a value for β_1 in the chain according to the following procedure:

Algorithm 8 *Step 1. Generate a candidate value* β_1' *from* $r\left(\beta_1|\beta_1^n\right)$.

Step 2. Fix $\beta_1^{n+1} = \beta_1'$ *with probability*

$$\gamma_M = \min\left\{\frac{L\left(Y|\beta_1'\right)\pi\left(\beta_1'\right)}{L\left(Y|\beta_1^n\right)\pi\left(\beta_1^n\right)}\frac{r\left(\beta_1^n|\beta_1'\right)}{r\left(\beta_1'|\beta_1^n\right)}, 1\right\}$$

and fix $\beta_1^{n+1} = \beta_1^n$ *with probability* $1 - \gamma_M$.

However, as the number of parameters in β_1 increases, the performance of the Metropolis algorithm worsens. The reason is that it is more difficult to find a proposal density that approximates the posterior well enough. A convenient way of generating new candidates is as in a random walk, that is, $\beta_1^{'} = \beta_1^n + v$, with v drawn from a symmetric distribution. If this scheme is chosen, the probability γ_M will not depend upon the function $r\left(.\right)$, but only on the ratio of the posterior evaluated at two different points. However, the larger the dimension of β_1 is, the more likely that new candidates fall into a region of small posterior probability, and hence get rejected. Hence, it is important to propose new values in a direction of high posterior probability.

Suppose that maximum likelihood estimates of β_1 are available. A normal or student-t proposal density $r(\beta_1)$ with mean $\widehat{\beta_1^{ML}}$ and covariance matrix $\widehat{\Delta}^{ML}$ equal to the maximum likelihood estimates, does not work well for relatively large dimensions. Alternatively, new candidates can be generated with a random walk $\beta_1^{'} = \beta_1^n + v$, with v having a variace-covariance matrix proportional to the maximum likelihood estimate. However, the algorithm could move faster by taking into account the information about the mode of the distribution.

The following Metropolis step, uses a random walk proposal density but it still uses the mode and estimated variance-covariance matrix of the distribution.

Algorithm 9 *Step 1. Generate a random vector v from a* $N\left(\widehat{\beta_1^{ML}}, \widehat{\Delta}^{ML}\right)$.

Step 2. Let $\overrightarrow{d} = \frac{(v-\beta_1^n)}{\|v-\beta_1^n\|} * \frac{(v_1-\beta_{11}^n)}{|(v_1-\beta_{11}^n)|}$, *where* v_1 *and* β_{11}^n *are the first elements in the vectors* v *and* β_1^n, *respectively.*

Step 3. Let $\beta_1' = \beta_1^n + c\overrightarrow{d}$, *where* c *is drawn from a* $N(0, \sigma_c)$

Step 4. Fix $\beta_1^{n+1} = \beta_1'$ *with probability*

$$\gamma_M = \min\left\{\frac{L(Y|\beta_1')\,\pi(\beta_1')}{L(Y|\beta_1^n)\,\pi(\beta_1^n)}, 1\right\}$$

and fix $\beta_1^{n+1} = \beta_1^n$ *with probability* $1 - \gamma_M$.

The algorithm is similar to the shoot-and-run algorithm (Chen and Schmeiser, 1993), and randomly chooses a unitary norm vector that determines the direction of movement for the new candidate. This unitary norm vector, $\overrightarrow{d}$, is chosen using a distribution centered in the maximum likelihood estimates. The length recovered along this direction is determined by c, and therefore by σ_c. The smaller σ_c, the nearer the new candidate will be from the old candidate and hence the more likely that it will be accepted. However, for the algorithm to move fast it is crucial that σ_c has a reasonably large value.

The next lemma proves that the proposal density is symmetric, and hence it cancels out in the probability γ_M.

Lemma 2 *In algorithm (9), the density function of new candidates is symmetric, that is,*

$$r(\beta_1'|\beta_1^n) = r\left(\beta_1^n|\beta_1^{'}\right)$$

Proof. The unitary norm vector $\overrightarrow{d}$ represents a direction in R^{k_1} and depends on $(k_1 - 1)$ components, since it can be written as:

$$\overrightarrow{d} = (1, d_1, d_2, ..., d_{k_1-1})^T \frac{1}{\left|\sqrt{1 + (d_1)^2 + (d_2)^2 + ... + (d_{k_1-1})^2}\right|}$$

Let d contain the random components of $\overrightarrow{d}$, that is, $d = (d_1, d_2, ..., d_{k_1-1})^T$. It is first proved that the density of (d, c) given β_1^n, $r(d, c|\beta_1^n)$, is the same as the density of $(d, -c)$ given $\beta_1^{'}$, $r\left(d, -c|\beta_1^{'}\right)$, when $\beta_1' = \beta_1^n + c\overrightarrow{d}$.

Since d and c are drawn independently, $r(d, c|\beta_1^n) = r(d|\beta_1^n)\,r(c|\beta_1^n)$, and $r\left(d, -c|\beta_1^{'}\right) = r\left(d|\beta_1^{'}\right) r\left(-c|\beta_1^{'}\right)$. Since c is drawn from a standard normal distribution and independently of $\left(\beta_1^n, \beta_1^{'}\right)$, then

$$\begin{aligned} r(d, c|\beta_1^n) &= r\left(d|\beta_1^{n}\right) r(c) \\ r\left(d, -c|\beta_1^{'}\right) &= r\left(d|\beta_1^{'}\right) r(c) \end{aligned}$$

Hence, $r(d, c|\beta_1^n) = r\left(d, -c|\beta_1^{'}\right)$ if and only if $r\left(d|\beta_1^{n}\right) = r\left(d|\beta_1^{'}\right)$, for $\beta_1' = \beta_1^n + c\overrightarrow{d}$.

Let d^* be the value of d that makes $\beta_1' = \beta_1^n + c\overrightarrow{d}$. Let $R\left(d^*|\beta_1^n\right)$ be the distribution function of d given β_1^n evaluated at d^*. Then

$$R\left(d^*|\beta_1^n\right) = \Pr\left\{\left(\frac{v_2-\beta_{12}^n}{v_1-\beta_{11}^n}, \frac{v_3-\beta_{13}^n}{v_1-\beta_{11}^n}, ..., \frac{v_{k_1}-\beta_{1k_1}^n}{v_1-\beta_{11}^n}\right) \leq \left(d_1^*, d_2^*, ..., d_{k_1-1}^*\right)\right\}$$

This probability is equal to:

$$R\left(d^*|\beta_1^n\right) = \Pr\left\{\begin{pmatrix} v_2 \\ v_3 \\ ... \\ v_{k_1} \end{pmatrix} \leq \begin{pmatrix} d_1^* \\ d_2^* \\ ... \\ d_{k_1-1}^* \end{pmatrix} v_1 + \begin{pmatrix} \beta_{12}^n \\ \beta_{13}^n \\ ... \\ \beta_{1k_1}^n \end{pmatrix} - \begin{pmatrix} d_1^* \\ d_2^* \\ ... \\ d_{k_1-1}^* \end{pmatrix} \beta_{11}^n\right\}$$

Similarly, the distribution function of d given $\beta_1^{'}$ evaluated at d^* is:

$$R\left(d^*|\beta_1^{'}\right) = \Pr\left\{\begin{pmatrix} v_2 \\ v_3 \\ ... \\ v_{k_1} \end{pmatrix} \leq \begin{pmatrix} d_1^* \\ d_2^* \\ ... \\ d_{k_1-1}^* \end{pmatrix} v_1 + \begin{pmatrix} \beta_{12}^{'} \\ \beta_{13}^{'} \\ ... \\ \beta_{1k_1}^{'} \end{pmatrix} - \begin{pmatrix} d_1^* \\ d_2^* \\ ... \\ d_{k_1-1}^* \end{pmatrix} \beta_{11}^{'}\right\}$$

From $\beta_1' = \beta_1^n + c\overrightarrow{d}$, it turns out that

$$\left(\beta_1^{'} - \beta_1^n\right)\frac{1}{c} = \overrightarrow{d}$$

Hence,

$$\left(\frac{\beta_{12}^{'}-\beta_{12}^n}{\beta_{11}'-\beta_{11}^n}, \frac{\beta_{13}^{'}-\beta_{13}^n}{\beta_{11}'-\beta_{11}^n}, ..., \frac{\beta_{1k_1}^{'}-\beta_{1k_1}^n}{\beta_{11}'-\beta_{11}^n}\right) = \left(d_1^*, d_2^*, ..., d_{k_1-1}^*\right)$$

and therefore:

$$\begin{pmatrix} \beta_{12}^{'} \\ \beta_{13}^{'} \\ ... \\ \beta_{1k_1}^{'} \end{pmatrix} - \begin{pmatrix} d_1^* \\ d_2^* \\ ... \\ d_{k_1-1}^* \end{pmatrix} \beta_{11}^{'} = \begin{pmatrix} \beta_{12}^n \\ \beta_{13}^n \\ ... \\ \beta_{1k_1}^n \end{pmatrix} - \begin{pmatrix} d_1^* \\ d_2^* \\ ... \\ d_{k_1-1}^* \end{pmatrix} \beta_{11}^n$$

Hence, $R\left(d^*|\beta_1^n\right) = R\left(d^*|\beta_1^{'}\right)$ and taking derivatives $r\left(d^*|\beta_1^n\right) = r\left(d^*|\beta_1^{'}\right)$, for all d^* such that $\beta_1' = \beta_1^n + c\overrightarrow{d}$.

Since $\beta_1' = \beta_1^n + c\overrightarrow{d}$, the density of $\beta_1^{'}$ given β_1^n is obtained from the density of $(d, c|\beta_1^n)$ by the theorem of change of variables. Similarly, since $\beta_1^n = \beta_1^{'} + (-c)\overrightarrow{d}$, the density of β_1^n given $\beta_1^{'}$ is obtained from the density of $(d, (-c)|\beta_1')$. Since $r\left(d, c|\beta_1^n\right) = r\left(d, (-c)|\beta_1^{'}\right)$ then $r\left(\beta_1'|\beta_1^n\right) = r\left(\beta_1^n|\beta_1^{'}\right)$. ■

6.2 The Multivariate Case.

In the case of the multivariate probit model of dimension T, an approximation of the slope parameters might be obtained by estimating by maximum likelihood T univariate probit models. Let $\widehat{\beta_i^{ML}}, \widehat{\Delta_i}^{ML}$ be the maximum likelihood estimates from running a simple probit for the ith equation of the multivariate probit model. If there is substantial prior information for the slope parameters, $\widehat{\beta_i^{ML}}, \widehat{\Delta_i}^{ML}$ would better be the mode and hessian of the log-posterior density for a univariate probit, which can also be obtained with a simple optimization routine. The grouping strategy in the algorithm is as follows:

- Sample Σ conditioning on $(Y_1^*, ..., Y_N^*)$ and $\{\beta_t : t = 1, ..., T\}$

- For $t = 1, ..., T$ do:
  - Generate β_t conditional on $\{y_{ik}^* : k \neq t\}^{i=1,...,N}$,$\{\beta_k : k \neq t\}$, Σ.
  - Sample $\{y_{it}^*\}^{i=1,...,N}$ conditional on $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \Sigma\right)$ and $\{y_{ik}^* : k \neq t\}^{i=1,...,N}$

The latent data is sampled from truncated normals, in the same way as in the above algorithms. Σ can be sampled using the algorithm in section 2. β_t can be generated adapting the algorithm above as follows:

Algorithm 10 *Step 1. Generate a random vector f from a $N\left(\sqrt{\sigma_{tt}}\widehat{\beta_t^{ML}}, D\widehat{\Delta_t}^{ML}D^T\right)$, where D is a diagonal matrix with all diagonal elements being equal to $\sqrt{\sigma_{tt}}$.*

Step 2. Let $\beta_t^l = \beta_t^n + c\frac{(f-\beta_t^n)}{\|f-\beta_t^n\|}$, where c is drawn from a $N(0, \sigma_t)$

Step 3. Fix $\beta_t^{n+1} = \beta_t^l$ with probability

$$\gamma_M = \min\left\{\frac{L_t\left(Y|\beta_t^l, \{\beta_k : k \neq t\}, \{y_{ik}^* : k \neq t\}^{i=1,...,N}, \Sigma\right)\pi\left(\beta_t^l\right)}{L_t\left(Y|\beta_t^n, \{\beta_k : k \neq t\}, \{y_{ik}^* : k \neq t\}^{i=1,...,N}, \Sigma\right)\pi\left(\beta_t^n\right)}, 1\right\}$$

and fix $\beta_t^{n+1} = \beta_t^n$ with $1 - \gamma_M$.

$L_t(.)$ is the likelihood of the model when we condition on the latent data for the other equations, excluding the kth equation.

7 Comparing the Performance of the Algorithms.

This section estimates a probit model and multivariate probit model in order to illustrate the distinctive characteristics of each algorithm. The next sub-section will focus on the probit model, and the multivariate case will be considered in section 7.2.

7.1 Univariate Probit Model.

Four algorithms will be compared: the standard Gibbs sampling algorithm, algorithm 2 (PX-DA), algorithm 5 and algorithm 7. Algorithm 5 is implemented with step 2' repeated 4 times. 8400 observations for seven explanatory variables were generated independently from a standard normal distribution. Slope coefficients are:

$$\beta_{11} = 1, \beta_{12} = 2, \beta_{13} = 0.5, \beta_{14} = -0.2, \beta_{15} = -1, \beta_{16} = 0.8, \beta_{17} = 0.8$$

Auto-correlations in the chain are calculated using 29000 iterations after discarding the first 1000 iterations. Parameters had an initial value equal to zero. Auto-correlations in algorithm 5 are the lowest, being less than half the correlations in algorithms 2 and 7. Auto-correlations in algorithm 2 are similar to those in algorithm 7, being both of them less than half of the auto-correlations in the Gibbs sampling algorithm. Table 1 shows the value of the highest correlation for lags 5, 10 and 20. In the Gibbs sampling algorithm, 80 lags are necessary for the autocorrelations of all parameters to be below 0.1. In algorithms 2 and 7 the same is achieved with 20 lags. Algorithm 5 needs the lowest amount of lags, 10, for all correlations to be below 0.1.

	Lag 5	Lag 10	Lag 20
Gibbs Algorithm	0.79	0.66	0.50
Algorithm 2	0.41	0.17	0.04
Algorithm 5	0.23	0.06	0.05
Algorithm 7	0.40	0.15	0.02

Table 1. Maximum Autocorrelation of the Parameters.

The highest correlation for all algorithms, except for algorithm 5, correspond to β_{12}. As noted by Liu and Wu (1999), autocorrelations in the probit model increase with the absolute value of the coefficients. In constrast, the autocorrelation of β_{12} in algorithm 5 is the lowest, and the highest correspond to parameter β_{14}.

Large auto-correlations make it more difficult to determine whether the chain has converged. With 29000 Gibbs sampling iterations, after discarding the first 1000 iterations, the Geweke test (1992) rejects the null hypothesis of convergence for 3 out of 7 parameters. With the same number of iterations, the test accepts the null hypothesis of convergence of all the parameters in the other three algorithms.

The Gibbs algorithm, algorithm 2 and algorithm 7 have similar computation time. However, algorithm 5 needs approximately double computing time, with this implementation. Hence, there is almost no benefit of using it compared to using algorithms 2 and 7, since similar value for correlations can be obtained with approximately the same computing time. However, as the following example shows, when slope parameters have a larger value the gains in autocorrelation clearly outweight the losses in computation time.

A similar excercise is carried out, with the same number of observations, but letting the value of the parameters be:

$$\beta_{11} = 3, \beta_{12} = 3, \beta_{13} = 3, \beta_{14} = -3, \beta_{15} = -3, \beta_{16} = -3, \beta_{17} = 3$$

Table 2 shows the maximum correlation of the parameters. Algorithm 2 needs at least 50 lags for correlations to be below 0.1. Algorithm 7 performs slightly better and needs 40 lags. Algorithm 5 have all correlations below 0.1 with just 5 lags. Hence, the substantial gains in smaller correlations in algorithm 5 more than compensate for the additional computation time per iteration.

	Lag 5	Lag 10	Lag 30	Lag 40	Lag 50
Algorithm 2	0.76	0.57	0.23	0.15	0.07
Algorithm 5	0.09	0.04	0.02	0.05	0.01
Algorithm 7	0.76	0.58	0.14	0.06	0.01

Table 2. Auto-correlations when parameters have a large value.

7.2 Multivariate Probit Model.

This section compares the performance of four algorithms: a Gibbs algorithm, Algorithm 4, Algorithm 6, and Algorithm 7. The data was generated according to the following random-effects type process:

$$y_{it}^* = 1 * x_{1i} + 2 * x_{2i} + 0.5 * x_{3i} - 0.2 * x_{4i} - 1 * x_{5i} + 0.8 * x_{6i} + 0.8 * x_{7i} + u_i + e_{it} \qquad i = 1, ..., 1200 \qquad t = 1, ..., 7$$

where e_{it} follows a $N(0, I)$, u_i follows a $N(0, 1)$, and it is independent of e_{it}. The regressors are invariant with t and are generated indepedently from a standard normal distribution. The prior for the slope parameters is a normal distribution with zero mean and covariance matrix equal to 10000I. The prior for the free parameters in Σ is a restricted inverted Wishart with $K_0 = I$ and $df_0 = 2 * T + 1 = 15$.

Hence, in this specification, 49 slope parameters plus 21 covariance parameters are estimated. For simplicity, only the 7 slope parameters in the first equation and 7 covariance parameters are analysed. Auto-correlations are calculated with 9000 iterations after discarding the first 1000.

Correlations for slope paremeters dissapear earliest in algorithm 6, at 40 lags. In algorithm 7, 50 lags are needed to make correlations dissapear. More than 50 lags are needed in Algorithm 4 and Gibbs algorithm for the correlations to be smaller than 0.1.

However, Table 4 shows that none of the algorithms considered succeeds in reducing Gibbs correlations for the covariance parameters. All algorithms considered have at least one parameter with a correlation as high as 0.13 after 100 lags. Since large autocorrelations make it more difficult to detect the convergence of the chain, Heidelberg et al (1983) test rejected the hypothesis of convergence in all algorithms with 9000 iterations after a burn-in period of 1000.

Hence, it seems necessary to consider an alternative algorithm that reduces the correlations and enhances the reliability of the estimations. For this purpose, consider the following parameterization of the multivariate

probit model.

$$
\begin{aligned}
\overline{\beta_t} &= \left(\beta_{t1}, \frac{\beta_{t2}}{\beta_{t1}}, \frac{\beta_{t3}}{\beta_{t1}}, ..., \frac{\beta_{tk_1}}{\beta_{t1}}\right) = \left(\beta_{t1}, \overline{\beta_{t2}}, ..., \overline{\beta_{tk_1}}\right) \qquad t = 1, ..., T \\
y_{it}^* &= X_{it1}\beta_{t1} + X_{it2}\overline{\beta_{t2}}\beta_{t1} + X_{it3}\overline{\beta_{t3}}\beta_{t1} + ... + X_{itk_t}\overline{\beta_{tk_t}}\beta_{t1} + e_{it} \qquad t = 1, ..., T \\
\overline{\Sigma_2} &= \left(\sigma_{12}/\left(\beta_{11}\beta_{21}\right), \sigma_{13}/\left(\beta_{11}\beta_{31}\right), ..., \sigma_{1T}/\left(\beta_{11}\beta_{T1}\right)\right) = \left(\overline{\sigma_{12}}, \overline{\sigma_{13}}, ..., \overline{\sigma_{1T}}\right) \\
\overline{\Sigma_3} &= \sigma_{23}/\left(\beta_{21}\beta_{31}\right), \sigma_{24}/\left(\beta_{21}\beta_{41}\right), ..., \sigma_{2T}/\left(\beta_{21}\beta_{T1}\right) = \left(\overline{\sigma_{23}}, \overline{\sigma_{24}}, ..., \overline{\sigma_{2T}}\right) \\
&... \\
\overline{\Sigma_T} &= \sigma_{(T-1)T}/\left(\beta_{T1}\beta_{(T-1)1}\right) = \overline{\sigma_{(T-1)T}}
\end{aligned}
$$

With this parameterization, the covariance matrix of (e_i) is equal to:

$$
\Sigma = \begin{pmatrix}
\sigma_{11} & \overline{\sigma_{12}}\beta_{11}\beta_{21} & ... & \overline{\sigma_{1T}}\beta_{11}\beta_{T1} \\
\overline{\sigma_{12}}\beta_{11}\beta_{21} & \sigma_{22} & ... & \overline{\sigma_{2T}}\beta_{21}\beta_{T1} \\
 & & ... & \\
\overline{\sigma_{1T}}\beta_{11}\beta_{T1} & \overline{\sigma_{2T}}\beta_{21}\beta_{T1} & ... & \sigma_{TT}
\end{pmatrix}
$$

where $(\sigma_{11}, \sigma_{22}, ..., \sigma_{TT})$ are determined by normalization (5) (e.g. $\sigma_{22} = 1 + \left(\overline{\sigma_{12}}\beta_{11}\beta_{21}\right)^2$). Note that normalization (5) ensures that matrix $\overline{\Sigma}$ is positive definite, and hence the covariance parameters are not subject to any restriction. One possibility to reduce correlations for covariance parameters is to draw some of their characteristics marginally on the latent data, as in the following algorithm:

Algorithm 11 *Step 1) Sample* $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}\right)$ *conditional on* $(Y_1^*, ..., Y_N^*, \Sigma)$.
Step 2) Sample Σ *conditional on* $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, Y_1^*, ..., Y_N^*\right)$
Step 3)For $t = 1, ..., T$ *do:*

- *Generate* β_{t1} *conditional on* $\{y_{ik}^* : k \neq t\}^{i=1,...,N}$,$\{\beta_{tk} : k \neq 1\}$, $\left\{\overline{\beta_j} : j \neq t\right\}$, $\left\{\overline{\Sigma_t} : t = 2, ..., T\right\}$.
- *Sample* $\{y_{it}^*\}^{i=1,...,N}$ *conditional on* $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \Sigma\right)$ *and* $\{y_{ik}^* : k \neq t\}^{i=1,...,N}$

The first part of step 3) updates the slope parameters and covariance terms jointly and marginally on the latent data. The other parts of the algorithm are carried out in the same way as in algorithm 6.

Two versions of this algorithm are implemented: in one the first part of step 3 is carried out 3 times (algorithm 11a), and in the other is carried out just once (algorithm 11b). As table 4 shows, the maximum correlation for covariance parameters vanishes before 30 and 40 lags in algorithms 11a and 11b. By contrast, the Gibbs algorithm needed 120 lags for the maximum correlation to be below 0.1. Correlations for slope parameters also diminish with respect to other algorithms, and show a similar pattern to the correlation for covariance parameters.

The computing time per iteration in Algorithm 11a and 11b is 2.6 and 1.9 times larger than in the Gibbs algorithm, respectively. The gains in lower correlations more than compensate for the extra computing time, since the number of iterations needed for the Gibbs correlations to be below 0.1 is about 4 and 3 times the number of iterations needed in algorithm 11a and 11b, respectively. In addition, the chains produced by algorithm 11a and 11b passed the Heidelberg test for all parameters, hence further increasing the reliability of the calculations.

	Lag 5	Lag 10	Lag 30	Lag 40	Lag 50
Gibbs Algorithm	0.72	0.57	0.28	0.20	0.13
Algorithm 4	0.68	0.51	0.17	0.11	0.12
Algorithm 6	0.67	0.51	0.14	0.06	0.02
Algorithm 7	0.68	0.51	0.17	0.12	0.04

Table 3. Auto-correlations for slope parameters.

	Lag 20	Lag 30	Lag 40	Lag 50	Lag 100
Gibbs Algorithm	0.40	0.34	0.29	0.23	0.13
Algorithm 4	0.44	0.39	0.33	0.27	0.13
Algorithm 6	0.44	0.38	0.32	0.26	0.14
Algorithm 7	0.41	0.32	0.25	0.21	0.13

Table 4. Auto-correlations for covariance parameters.

	Lag 10	Lag 20	Lag 30	Lag 40	Lag 50
Gibbs Algorithm	0.54	0.40	0.34	0.29	0.23
Algorithm 11a	0.29	0.13	0.08	0.04	0.04
Algorithm 11b	0.44	0.23	0.12	0.08	0.06

Table 5. Auto-correlations for covariance parameters.

	Lag 10	Lag 20	Lag 30	Lag 40	Lag 50
Gibbs Algorithm	0.57	0.37	0.28	0.20	0.13
Algorithm 11a	0.33	0.17	0.09	0.06	0.04
Algorithm 11b	0.43	0.24	0.15	0.09	0.04

Table 6. Auto-correlations for slope parameters.

8 Discussion.

The motivation underlying the algorithms proposed in this paper is that a pure Gibbs algorithm moves slowly due to sampling separately variables that are highly correlated. The PX-DA algorithm was re-interpreted as a re-parameterization, in which some characteristics of the latent data are sampled marginally on slope parameters.

This new interpretation has several advantages. Firstly, it avoids the need of finding the Haar prior for the non-identified parameter. Secondly, it simplifies the understanding of the algorithm by identifying it with a class of algorithms that already existed (i.e. Gibbs and Metropolis algorithms). In particular, it avoids the interpretation of the algorithm as the limit of the kernel of a Gibbs sampling algorithm (Liu and Wu 1999). This, for instance, simplifies the justification to use a Metropolis step. Thirdly, this interpretation suggest re-parameterizations that yield faster algorithms. In particular, as the previous section shows, sampling some characteristics of the slope parameters marginally on the latent data (algorithm 4) proves to be a better parameterization in the probit model. In the multivariate probit model, it seems necessary to sample not only characteristics of the slope parameters but also of the covariance parameters marginally on the latent data (algorithm 11).

Section 5 of this paper presented an algorithm in which some characteristics of latent data and slope parameters were updated jointly. For the univariate probit, it has the advantage with respect to the PX-DA algorithm that conditional distributions can be sampled directly even if a proper prior for the slope parameters is used. In addition, the previous section showed that the performance of this algorithm can be slightly better than the PX-DA algorithm, for the same computation time. These two algorithms outperform the Gibbs sampling algorithm since they achieve lower correlations with virtually the same computation time. Section 7 showed that the extra time per iteration required in algorithm 4 is sometimes small compared to the gains in lower correlations, specially in probit models with large posterior standard deviations.

Section 2 proposes an alternative normalization that allows direct sampling of the covariance matrix. Sampling directly speeds up convergence compared to a Metropolis algorithm, which only asymptotically obtains a draw from the conditional distribution. In addition, it requires less computation time, and it avoids finding a proposal density, which often do not work well when there are many covariance parameters.

By choosing an alternative normalization, the conditional density of the free parameters in Σ is known (appendix 1). This substantially simplifies the calculation of the predictive density, needed for model selection, and allows the application of Chib (1995) method. Notwithstanding, the parameters in the chain can be transformed to obtain results according to a different normalization (section 2.3).

When the number of parameters is large, a Metropolis step rarely works well. If a pure random walk is used, new candidates are very likely to fall in a region of low probability and hence get rejected. On the other hand, a proposal density that is an approximation of the conditional posterior is usually not good enough, especially in the tails of the distribution, which makes new candidates often rejected. Section 7 presented a random walk proposal density which moves in directions of high probability. Since it is a random walk, the proposal density cancels out in the acceptance probability, and hence the approximation problem is avoided. This Metropolis algorithm might be useful when some information about the mode and dispersion of the posterior distribution is obtained, for instance, from a preliminary highly correlated Markov Chain.

Improper priors are sometimes computationally more convenient, allowing direct sampling from the conditional distributions (proposition 7 in appendix 3, and section 3.2.1). However, more research is needed to know whether the posterior distribution is proper when these priors are used.

The type of re-parameterizations considered in this paper make it possible to update large numbers of parameters jointly and marginally on the latent data. This is potentially applicable to many models with complex likelihooods, where conventional MCMC algorithms fail to yield reliable calculations in reasonable time.

Acknowledgement: I thank Karim Abadir, Giovanni Forchini, Andrew Jones and Nigel Rice for useful comments and suggestions.

Appendix 1.

Let $f_N(x;\mu,\Phi)$ denote the density function of a $N(\mu,\Phi)$, evaluated at x. And let $f_{IW}(x;p,df,K)$ denote the density function of an inverted Wishart of dimension p, degrees of freedom df, and expected value $K\frac{1}{df-2p-2}$.

Proposition 4 *If Σ follows an unrestricted inverted Wishart $IW(T,df,K)$, then the density function of $\left(\sigma_{11},\sigma_{22\cdot1},\sigma_{12},\sigma_{33\cdot12},\Sigma_3,...,\sigma_{TT\cdot12...(T-1)},\Sigma_T\right)$ can be expressed as:*

$$
\begin{aligned}
&f_{IW}(\sigma_{11};1,df-2T+2,k_{11})\times\\
&f_{IW}(\sigma_{22\cdot1};1,df-2T+3,k_{22\cdot1})\times\\
&f_N\left(\sigma_{12};\sigma_{11}(k_{11})^{-1}K_2,\sigma_{11}(k_{11})^{-1}\sigma_{11}\right)\times\\
&f_{IW}(\sigma_{33\cdot12};1,df-2T+4,k_{33\cdot12})\times\\
&f_N\left(\Sigma_3;\Sigma_{22}(K_{22})^{-1}K_3,\Sigma_{22}(K_{22})^{-1}\Sigma_{22}\right)\times\\
&...\times\\
&...\times\\
&f_{IW}\left(\sigma_{TT\cdot12...(T-1)};1,df-2T+(T+1),k_{TT\cdot12...(T-1)}\right)\times\\
&f_N\left(\Sigma_T;\Sigma_{(T-1)(T-1)}\left(K_{(T-1)(T-1)}\right)^{-1}K_T,\Sigma_{(T-1)(T-1)}\left(K_{(T-1)(T-1)}\right)^{-1}\Sigma_{(T-1)(T-1)}\right)
\end{aligned}
$$

Proof. This decomposition results from applying recursively Theorem 1. ■

Appendix 2.

Let Y^* be a $T\times T$ diagonal matrix, with diagonal equal to $(y^*_{11},y^*_{12},...,y^*_{1T})$. The following theorem gives the conditional density of $\Sigma|Y^*,\beta,\left\{\left(\overline{y^*_{2j}},\overline{y^*_{3j}},...,\overline{y^*_{Nj}}\right):j=1,...,T\right\}$ and the density of $Y^*|\beta,\left\{\left(\overline{y^*_{2j}},\overline{y^*_{3j}},...,\overline{y^*_{Nj}}\right):j=1,...,T\right\}$.

Proposition 5 *The conditional posterior density of Σ given $\left(Y^*,\beta,\left\{\left(\overline{y^*_{2j}},\overline{y^*_{3j}},...,\overline{y^*_{Nj}}\right):j=1,...,T\right\}\right)$ is proportional to:*

$$f_{IW}\left(\Sigma;T,df,\widetilde{K}\right)$$

with $df=df_0+N$, $\widetilde{K}=K_0+\overline{K}$, and $\overline{K}=\sum_{i=1}^{N}\left(\widetilde{e}_i\widetilde{e}_i^T\right)$, and

$$\widetilde{e}_1=\begin{pmatrix}y^*_{11}-X^T_{11}\beta_1\\ y^*_{12}-X^T_{12}\beta_2\\ ...\\ y^*_{1T}-X^T_{1T}\beta_T\end{pmatrix}\quad\text{and }\widetilde{e}_i=\begin{pmatrix}y^*_{11}\overline{y^*_{i1}}-X^T_{i1}\beta_1\\ y^*_{12}\overline{y^*_{i2}}-X^T_{i2}\beta_2\\ ...\\ y^*_{1T}\overline{y^*_{iT}}-X^T_{iT}\beta_T\end{pmatrix}$$

The conditional posterior of Y^* given $\left(\beta,\left\{\left(\overline{y^*_{2j}},\overline{y^*_{3j}},...,\overline{y^*_{Nj}}\right):j=1,...,T\right\}\right)$ is proportional to:

$$
\begin{aligned}
&|Y^*|^{N-1}\left|\widetilde{K}\right|^{-(df-T-1)/2}f_{IW}\left(1;1,df-2T+2,\widetilde{k_{11}}\right)\times f_{IW}\left(1;1,df-2T+3,\widetilde{k_{22\cdot1}}\right)\\
&\times f_{IW}\left(1;1,df-2T+4,\widetilde{k_{33\cdot12}}\right)\times f_{IW}\left(1;1,df-2T+4,\widetilde{k_{33\cdot12}}\right)\times...\\
&\times f_{IW}\left(1;1,df-2T+(T+1),\widetilde{k_{TT\cdot12...(T-1)}}\right)
\end{aligned}
\tag{10}
$$

Proof. The conditional posterior density of (Y^*,Σ) given $\left(\beta,\left\{\left(\overline{y^*_{2j}},\overline{y^*_{3j}},...,\overline{y^*_{Nj}}\right):j=1,...,T\right\}\right)$ is proportional to:

$$|Y^*|^{N-1}f_{IW}(\Sigma;T,df_0,K_0)|\Sigma|^{-N/2}\exp\left\{-\frac{1}{2}tr\left(\Sigma^{-1}\overline{K}\right)\right\}\tag{11}$$

where $f_{IW}(\Sigma; T, df_0, K_0)$ is the prior density for Σ.

This expression is proportional to:

$$|Y^*|^{N-1} \frac{\left|\widetilde{K}\right|^{(df-T-1)/2}}{\left|\widetilde{K}\right|^{(df-T-1)/2}} |\Sigma|^{-(N+df_0)/2} \exp\left\{-\frac{1}{2} tr\left(\Sigma^{-1}\widetilde{K}\right)\right\}$$

This last expression is proportional to:

$$|Y^*|^{N-1} \frac{1}{\left|\widetilde{K}\right|^{(df-T-1)/2}} f_{IW}\left(\Sigma; T, df, \widetilde{K}\right) \tag{12}$$

Decomposing $f_{IW}\left(\Sigma; T, df, \widetilde{K}\right)$ into the product given by proposition (4), and integrating the free elements in Σ gives the result. ■

It follows from this proposition that $\Sigma|Y^*, \beta, \left\{\left(\overline{y^*_{2j}}, \overline{y^*_{3j}}, ..., \overline{y^*_{Nj}}\right) : j = 1, ..., T\right\}$ can be sampled directly using algorithm (1). Y^* could be generated using a Metropolis step, with a random walk as proposal density.

Appendix 3.

Let Y^* be a $T \times T$ diagonal matrix, with diagonal equal to $(y^*_{11}, y^*_{12}, ..., y^*_{1T})$. The following theorem gives the conditional density of $\Sigma|Y^*, \overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \left\{\left(\overline{y^*_{2j}}, \overline{y^*_{3j}}, ..., \overline{y^*_{Nj}}\right) : j = 1, ..., T\right\}$ and the density of $Y^*|\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \left\{\left(\overline{y^*_{2j}}, \overline{y^*_{3j}}, ..., \overline{y^*_{Nj}}\right) : j = 1, ..., T\right\}$. Let $k = k_1 + k_2 + ... + k_T$.

Proposition 6 *The conditional posterior of* $(y^*_{11}, y^*_{12}, ..., y^*_{1T})$ *given* $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \left\{\left(\overline{y^*_{2j}}, \overline{y^*_{3j}}, ..., \overline{y^*_{Nj}}\right) : j = 1, ..., T\right\}\right)$ *is proportional to:*

$$\begin{aligned} &|Y^*|^{N+k-1} \left|\widetilde{K}\right|^{-(df-T-1)/2} f_{IW}\left(1; 1, df - 2T + 2, \widetilde{k_{11}}\right) \times f_{IW}\left(1; 1, df - 2T + 3, \widetilde{k_{22\cdot 1}}\right) \\ &\times f_{IW}\left(1; 1, df - 2T + 4, \widetilde{k_{33\cdot 12}}\right) \times f_{IW}\left(1; 1, df - 2T + 4, \widetilde{k_{33\cdot 12}}\right) \times ... \\ &\times f_{IW}\left(1; 1, df - 2T + (T+1), \widetilde{k_{TT\cdot 12...(T-1)}}\right) \end{aligned} \tag{13}$$

where

$$\widetilde{e}_1 = \begin{pmatrix} 1 - X^T_{11}\overline{\beta_1} \\ 1 - X^T_{12}\overline{\beta_2} \\ ... \\ 1 - X^T_{1T}\overline{\beta_T} \end{pmatrix} \quad \text{and } \widetilde{e}_i = \begin{pmatrix} \overline{y^*_{i1}} - X^T_{i1}\overline{\beta_1} \\ \overline{y^*_{i2}} - X^T_{i2}\overline{\beta_2} \\ ... \\ \overline{y^*_{iT}} - X^T_{iT}\overline{\beta_T} \end{pmatrix} \quad \text{for } i = 2, ..., N$$

and $\widetilde{K} = Y^* K Y^* + K_0$, $K = \sum_{i=1}^N \left(\widetilde{e}_i \widetilde{e}_i^T\right)$.

Proof. The conditional posterior of (Y^*, Σ) is proportional to:

$$\exp\left\{-\frac{1}{2} tr\left(\Sigma^{-1} K_0\right)\right\} |\Sigma|^{-(N+df_0)/2} |Y^*|^{N+k-1} \exp\left\{-\frac{1}{2} \sum_{i=1}^N \left(Y^* \widetilde{e}_i\right)^T \Sigma^{-1} \left(Y^* \widetilde{e}_i\right)\right\} \tag{14}$$

Operating, this expression is equal to:

$$\exp\left\{-\frac{1}{2} tr\left(\Sigma^{-1} K_0\right)\right\} |Y^*|^{N+k-1} |\Sigma|^{-(N+df_0)/2} \exp\left\{-\frac{1}{2} tr\left(Y^* \Sigma^{-1} Y^* K\right)\right\} \tag{15}$$

with $\widetilde{K} = \sum_{i=1}^{N} (\widetilde{e}_i)^T (\widetilde{e}_i) + K_0$. This expression is equal to:

$$\frac{\left|\widetilde{K}\right|^{(df-T-1)/2}}{\left|\widetilde{K}\right|^{(df-T-1)/2}} |\Sigma|^{-(N+df_0)/2} |Y^*|^{N+k-1} \exp\left\{-\frac{1}{2} tr\left(\Sigma^{-1}\left(Y^* K Y^* + K_0\right)\right)\right\} \tag{16}$$

$$|Y^*|^{N+k-1} \frac{1}{\left|\widetilde{K}\right|^{(df-T-1)/2}} f_{IW}\left(\Sigma; T, df, \widetilde{K}\right) \tag{17}$$

Decomposing $f_{IW}\left(\Sigma; T, df, \widetilde{K}\right)$ into the product given by proposition (4), and integrating the free elements in Σ gives the result. ■

The next proposition shows that if the improper prior $\pi(\Sigma) \propto 1$ is used for the free elements in Σ, then (Y^*, Σ) can be sampled directly.

Proposition 7 *Let the prior for Σ be $\pi(\Sigma) \propto 1$. The conditional posterior of $\left((y_{11}^*)^{-2}, (y_{12}^*)^{-2}, ..., (y_{1T}^*)^{-2}\right)$ given $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, \left\{\left(\overline{y_{2j}^*}, \overline{y_{3j}^*}, ..., \overline{y_{Nj}^*}\right) : j = 1, ..., T\right\}\right)$ is proportional to:*

$$\begin{aligned} &f_{IW}\left(\left(\overline{y_{11}^*}\right)^{-2}; 1, df - T + k + 3, k_{11}\right) \times f_{IW}\left(\left(\overline{y_{12}^*}\right)^{-2}; 1, df - T + k + 4, k_{22\cdot 1}\right) \\ &\times f_{IW}\left(\left(\overline{y_{13}^*}\right)^{-2}; 1, df - T + k + 5, k_{33\cdot 12}\right) \times f_{IW}\left(\left(\overline{y_{14}^*}\right)^{-2}; 1, df - T + k + 6, k_{33\cdot 12}\right) \times ... \\ &\times f_{IW}\left(\left(\overline{y_{1T}^*}\right)^{-2}; 1, df - T + k + (T+2), k_{TT\cdot 12...(T-1)}\right) \end{aligned} \tag{18}$$

where

$$\widetilde{e}_1 = \begin{pmatrix} 1 - X_{11}^T \overline{\beta_1} \\ 1 - X_{12}^T \overline{\beta_2} \\ ... \\ 1 - X_{1T}^T \overline{\beta_T} \end{pmatrix} \quad \text{and } \widetilde{e}_i = \begin{pmatrix} \overline{y_{i1}^*} - X_{i1}^T \overline{\beta_1} \\ \overline{y_{i2}^*} - X_{i2}^T \overline{\beta_2} \\ ... \\ \overline{y_{iT}^*} - X_{iT}^T \overline{\beta_T} \end{pmatrix} \quad \text{for } i = 2, ..., N$$

$K = \sum_{i=1}^{N} \left(\widetilde{e}_i \widetilde{e}_i^T\right)$, *and* $df = N$.

Proof. Proposition can be applied to this case, noting that $\widetilde{K} = Y^* K Y^*$. Since $\widetilde{K} = Y^* K Y^*$, expresion (14) can be writen as:

$$\begin{aligned} &|Y^*|^{(k+T)} f_{IW}\left(\left(y_{11}^*\right)^{-2}; 1, df - 2T, k_{11}\right) \times f_{IW}\left(\left(y_{12}^*\right)^{-2}; 1, df - 2T + 1, k_{22\cdot 1}\right) \\ &\times f_{IW}\left(\left(y_{13}^*\right)^{-2}; 1, df - 2T + 2, k_{33\cdot 12}\right) \times f_{IW}\left(\left(y_{14}^*\right)^{-2}; 1, df - 2T + 4, k_{33\cdot 12}\right) \times ... \\ &\times f_{IW}\left(\left(y_{1T}^*\right)^{-2}; 1, df - 2T + (T-1), k_{TT\cdot 12...(T-1)}\right) \end{aligned} \tag{19}$$

Adjusting the degrees of freedom, this expression becomes:

$$\begin{aligned} &f_{IW}\left(\left(y_{11}^*\right)^{-2}; 1, df - T + k, k_{11}\right) \times f_{IW}\left(\left(y_{12}^*\right)^{-2}; 1, df - T + k + 1, k_{22\cdot 1}\right) \\ &\times f_{IW}\left(\left(y_{13}^*\right)^{-2}; 1, df - T + k + 2, k_{33\cdot 12}\right) \times f_{IW}\left(\left(y_{14}^*\right)^{-2}; 1, df - T + k + 3, k_{33\cdot 12}\right) \times ... \\ &\times f_{IW}\left(\left(y_{1T}^*\right)^{-2}; 1, df - T + k + (T-1), k_{TT\cdot 12...(T-1)}\right) \end{aligned} \tag{20}$$

The density of $\left((y_{11}^*)^{-2}, (y_{12}^*)^{-2}, ..., (y_{1T}^*)^{-2}\right)$ is obtained multiplying by the Jacobian and is equal to:

$$\begin{aligned} & f_{IW}\left((y_{11}^*)^{-2}; 1, df - T + k + 3, k_{11}\right) \times f_{IW}\left((y_{12}^*)^{-2}; 1, df - T + k + 4, k_{22\cdot 1}\right) \\ & \times f_{IW}\left((y_{13}^*)^{-2}; 1, df - T + k + 5, k_{33\cdot 12}\right) \times f_{IW}\left((y_{14}^*)^{-2}; 1, df - T + k + 6, k_{33\cdot 12}\right) \times ... \\ & \times f_{IW}\left((y_{1T}^*)^{-2}; 1, df - T + k + (T+2), k_{TT\cdot 12...(T-1)}\right) \end{aligned} \tag{21}$$

■

Appendix 4.

Proof of lemma 1.

Proof. If ε_i follows a $N(0, I)$, then the linear combination $\Delta\varepsilon_i$ also follows a normal $N(0, \Sigma)$. Hence, e_i can be expressed as $e_i = \Delta\varepsilon_i$, that is:

$$\begin{aligned} e_{i1} &= \delta_{11}\varepsilon_{i1} \\ e_{i2} &= \delta_{21}\varepsilon_{i1} + \delta_{22}\varepsilon_{i2} \\ e_{i3} &= \delta_{31}\varepsilon_{i1} + \delta_{32}\varepsilon_{i2} + \delta_{33}\varepsilon_{i3} \\ e_{i4} &= \delta_{41}\varepsilon_{i1} + \delta_{42}\varepsilon_{i2} + \delta_{43}\varepsilon_{i3} + \delta_{44}\varepsilon_{i4} \\ &... \\ e_{iT} &= \delta_{T1}\varepsilon_{i1} + \delta_{T2}\varepsilon_{i2} + \delta_{T3}\varepsilon_{i3} + \delta_{T4}\varepsilon_{i4} + ... + \delta_{TT}\varepsilon_{iT} \end{aligned} \tag{22}$$

Using these equations, Σ can be related to Δ:

$Var(e_{i1}) = \sigma_{11} = Var(\delta_{11}\varepsilon_{i1}) = (\delta_{11})^2$
$Cov(e_{i1}, e_{i2}) = \sigma_{12} = Cov(\delta_{11}\varepsilon_{i1}, \delta_{21}\varepsilon_{i1} + \delta_{22}\varepsilon_{i2}) = \delta_{11}\delta_{21}$
$Var(e_{i2}|e_{i1}) = \sigma_{22\cdot 1} = Var(\delta_{21}\varepsilon_{i1} + \delta_{22}\varepsilon_{i2}|\delta_{11}\varepsilon_{i1}) = (\delta_{22})^2$
$Cov(e_{i3}, e_{i1}) = \delta_{31}\delta_{11}$, $Cov(e_{i3}, e_{i2}|e_{i1}) = \delta_{32}\delta_{22}$, $Var(e_{i3}|e_{i1}, e_{i2}) = (\delta_{33})^2$
$Cov(e_{i4}, e_{i1}) = \delta_{41}\delta_{11}$, $Cov(e_{i4}, e_{i2}|e_{i1}) = \delta_{42}\delta_{22}$, $Cov(e_{i4}, e_{i3}|e_{i1}, e_{i2}) = \delta_{43}\delta_{33}$, $Var(e_{i4}|e_{i1}, e_{i2}, e_{i3}) = (\delta_{44})^2$
$Cov(e_{iT}|e_{i1}) = \delta_{T1}\delta_{11}$, $Cov(e_{iT}, e_{i2}|e_{i1}) = \delta_{T2}\delta_{22}$, $Cov(e_{iT}, e_{i3}|e_{i1}, e_{i2}) = \delta_{T3}\delta_{33}$,...,
$Var\left(e_{iT}|e_{i1}, e_{i2}, e_{i3}, ..., e_{i(T-1)}\right) = (\delta_{TT})^2$

■

References:

Amit, Y. (1991) "On rates of convergence of stochastic relaxation for Gaussian and non-Gaussian distributions" Journal of Multivariate Analysis., 38, 82-99.

Bauwens, L., Lubrano, M. and Richard, J. (1999) *Bayesian Inference in Dynamic Econometrics Models.* Oxford University Press, Oxford.

Chen, M. H. and Schmeiser, B.W. (1993) Performances of the Gibbs, hit-and-run, and Metropolis samplers. J. Comp. Graph Statist. 2, 251-272.

Chen M. H., and Shao (2001), "Propriety of Posterior Distribution for Dichotomous Quantal Response Models" Proceedings of the American Mathematical Society. Vol. 129, Number 1, pp. 293-302. Available at http://www.ams.org/proc/2001-129-01/

Chib, S. (1995), "Marginal Likelihood from the Gibbs Output" *Journal of the American Statistical Association* 90, 432, pp. 1313-21.

Chib, S. and Greenberg, E. (1998) Analysis of multivariate probit models. *Biometrika*, 85:347-361.

Cowles, K., Carlin, B. P., and Connett, J. E. (1996) "Bayesian Tobit Modellig of Longitudinal Ordinal Clinical Trial Compliance Data with Nonignorable Missingness," *Journal of the American Statistical Association*, 91, 86-98.

Gelfand, A. E., and Smith, A. F. M. (1990) "Sampling Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, 85, 398-409.

Geweke, J. (1992) "Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments," in *Bayesian Statistics (*Vol. *4)*, eds. J. M. Bernardo, J. Berger, A. P. Dawid, and A. F. M. Smith, Oxford:Oxford University Press, pp. 169-193.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., (1995) "Introducing Markov chain Monte Carlo" In Markov Chain Monte Carlo in Practice, (ed. W.R. Gilks, S. Richardson and D.j. Spiegelhalter), pp. 1-19.

Gilks, W. R., and Roberts, G. O. (1995), "Strategies for improving MCMC" In Markov Chain Monte Carlo in Practice, (ed. W.R. Gilks, S. Richardson and D.j. Spiegelhalter), pp. 89-114.

Hastings, W. K. (1970), Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57, 97-109.

Heidelberg, P., and Welch, P. D. (1983) "Simulation Run Length Control in the Presence of and Initial Transient," *Operations Research*, 31, 1109-1144.

Liu, C., (2001) "The Art of Data Augmentation: Discussion", *Journal of Computational and Graphical Statistics*, 10, 1, pp. 75-81.

Liu, J. S., and Sabatti, C. (1998), "Simulated Sintering: Markov Chain Monte Carlo With Spaces of Varying Dimensions" (with discussion), in Bayesian Satistics 6, eds. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, New York:Oxford University Press, pp.389-413.

Liu, J. S., and Sabatti, C. (2000), "Generalised Gibbs Sampler and Multigrid Monte Carlo for Bayesian Computation", *Biometrika*, 87, 2, 353-369.

Liu, J. S., and Wu, Y. N., (1999) "Parameter Expansion for Data Augmentation", *Journal of the American Statistical Association*, 94, 448, 1264-1274.

McCulloch, R. E., Polson, N. G., and Rossi, P. E., (2000) A Bayesian Analysis of the Multinomial Probit Model with Fully Identified Parameters. *Journal of Econometrics* 99: 173-193.

Meng, X. L., and van Dyk, D. (1999), "Seeking Efficient Data Augmentation Schemes via Conditional and Marginal Augmentation," *Biometrika*, 86, 301-320.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953), Equation of state calculations by fast compuing machine. *Journal of Chemical Physics*, 21:1087-91.

Natarajan, R., and McCulloch, C. E. (1995) "A Note on the Existence of the Posterior Distribution for a Class of Mixed Models for Binomial Responses," *Biometrika*, 82, 639-643.

Nobile, A.(2000), "Comment: Bayesian Multinomial Probit Models with a Normalization Constraint" *Journal of Econometrics*, 99, 2, pp. 335-45.

Philippe, A., and Robert, C. P., (2001) "Perfect simulation of positive Gaussian distributions." Available at http://www.ceremade.dauphine.fr/CMD/preprints01/0129perfect.pdf

Press, S. J. (1982) *Applied Multivariate Analysis: Using Bayesian and Frequentist Methods of Inference*, Krieger.

Raftery, A. L. and Lewis, S. (1992). How many iterations in the Gibbs sampler? In Bayesian Statistics 4, (ed. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith), pp.763-74. Oxford University Press.

Roberts, G. O., (1995) "Markov chain concepts related to sampling algorithms" In Markov Chain Monte Carlo in Practice, (ed. W.R. Gilks, S. Richardson and D.j. Spiegelhalter), pp. 45-57.

Tanner, T. A., and Wong, W. H. (1987) "The Calculation of Posterior Distributions by Data Augmentation," *Journal of the American Statistical Association*, 82, 528-550.

Tierney, L. (1994) "Markov Chains for Exploring Posterior Distributions (with discussion)" *Annals of Statistics*, 22, 1701-1762.

van Dyk, D. A., and Meng X. L, (2001) "The Art of Data Augmentation (with discussion)" *Journal of Computational and Graphical Statistics*, 10, 1, pp. 1-111.