New Matrix-Based Methods for the Analytic Evaluation of the Multivariate Cumulative Normal Distribution Function

Chandra R. Bhat (corresponding author)
The University of Texas at Austin
Department of Civil, Architectural and Environmental Engineering
301 E. Dean Keeton St. Stop C1761, Austin TX 78712, USA
Tel: 1-512-471-4535; Email: bhat@mail.utexas.edu
and
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ABSTRACT

In this paper, we develop a new matrix-based implementation of the Mendell and Elston (ME) analytic approximation to evaluate the multivariate normal cumulative distribution (MVNCD) function, using an LDLT decomposition method followed by a rank 1 update of the LDLT factorization. Our implementation is easy to code for individuals familiar with matrix-based coding. Further, our new matrix-based implementation for the ME algorithm allows us to efficiently write the analytic matrix-based gradients of the approximated MVNCD function with respect to the abscissae and correlation parameters, an issue that is important in econometric model estimation. In addition, we propose four new analytic methods for approximating the MVNCD function. The paper then evaluates the ability of the multiple approximations for individual MVNCD evaluations as well as multinomial probit model estimation. As expected, in our tests for evaluating individual MVNCD functions, we found that the traditional GHK approach degrades rapidly as the dimensionality of integration increases. Concomitant with this degradation in accuracy is a rapid increase in computational time. The analytic approximation methods are also much more stable across different numbers of dimensions of integration, and even the simplest of these methods is superior to the GHK-500 beyond seven dimensions of integration. Based on all the evaluation results in this paper, we recommend the new Two-Variate Bivariate Screening (TVBS) method proposed in this paper as the evaluation approach for MVNCD function evaluation.

Keywords: multivariate normal cumulative distribution function, multinomial probit, discrete choice models, econometric models.

1. INTRODUCTION

Many statistical and econometric applications require the evaluation of the multivariate normal cumulative distribution (MVNCD) function. For example, in consumer choice analysis in general, and transportation and marketing analysis in particular, the estimation of such models as the multinomial probit (MNP) model, the multivariate binary and ordered-response models, and the multiple discrete-continuous model all require the computation of the MVNCD function. The computation of the MVNCD function also features in applications in a variety of other stochastic programming fields such as defense and environmental economics, geography, and water management.

Earlier studies have developed many ways to evaluate the MVNCD function. While these methods are general and can be used in a variety of situations, most earlier studies have examined the MVNCD function evaluation in the context of MNP estimation. Some of these earlier MVNCD evaluations are based on simulation techniques and others on analytic approximations. Among the simulation methods, the best known approach within a frequentist estimation framework is the GHK probability simulator, named after Geweke (1991), Hajivassiliou (Hajivassiliou and McFadden, 1998), and Keane (1990, 1994). The GHK approach starts with transforming the *correlated* random terms into linear functions of *uncorrelated* standard normal deviates using the Cholesky decomposition of the correlation matrix in the MVNCD evaluation. Doing so helps in recasting the MVNCD as a recursive product of univariate (conditional) cumulative normal distributions (UCCNCD). Each UCCNCD involves the integral over a single-sided truncated normal, which is achieved in the GHK through simulation of pseudo-random draws from a truncated normal. Bhat et al. (2010) embed the Halton approach (rather than the pseudo-random approach) to draw from the truncated normal, because of the better coverage of the Halton draws over the integration space (see Bhat, 2001). Alternative GHK procedures involve the use of multivariate quadrature using sparse grid integration (SGI) (see Heiss and Winschel, 2008) or the use of Efficient Importance Sampling (EIS) within the GHK simulator (see Heiss, 2010). In addition to the above frequentist approach, efficient MCMC methods for models involving the MVNCD function evaluation have also been proposed through data augmentation techniques that make it possible to use standard Bayesian regression techniques (see McCulloch et al., 2000 and Imai and van Dyk, 2005).

Among the analytic approximation techniques, one of the first approaches was that proposed by Clark (1961). Unfortunately, this approximation does not perform well for MVNCD evaluations when the random variables are highly correlated or have different variances. In another analytic approximation study, Mendell and Elston (1974) (ME) use the same univariate conditioning approach that formed the basis later for the GHK, except they replace draws from the truncated normal at each conditioning step with approximations of the first two moments of the truncated variables at earlier conditioning steps. This method has also been used by many other authors since, including Rice et al. (1979), Kamakura (1989), and Hutmacher and French (2011). Yet another MVNCD analytic approximation was first proposed by Solow (1990) based on Switzer (1977), and then refined by Joe (1995). This procedure entails the decomposition of the multivariate integral into a product of conditional probabilities. At each step, the conditional probability is approximated based on replacing the conditional events by binary variables and the conditional probability itself as an expectation in a linear regression (with the binary variables for the conditional events serving as exogenous variables, with known covariances amongst themselves based on the correlation matrix of the MVNCD evaluation).

With the many simulation and analytic approaches, a couple of recent studies have examined the accuracy and precision offered by the many approaches in the context of MNP model estimation. Patil et al. (2017) compared all the simulation techniques mentioned earlier with Bhat's (2011) maximum approximate composite marginal likelihood (MACML) approach, which combines the Switzer-Solow-Joe (SSJ) approximation for the MVNCD function with the composite marginal likelihood (CML) inference approach for MNP models. The focus of Patil et al.'s (2017) study was on the accuracy and precision of MNP parameter recovery in a five-alternative choice context. They find that, among all the simulation-based techniques, the GHK-Halton performs best for MNP estimation. However, among the simulation and MACML approaches, the overall winner in terms of accuracy and precision of underlying parameter recovery, as well as computational time, is the MACML procedure with but one permutation of the ordering of the random term abscissae in its embedded analytic approximation for the MVNCD function. Connors et al. (2014) focused on the analytic approximations corresponding to the ME method and the SSJ method, though they also included the GHK and a couple of other simulation approaches for reference reasons. Unlike Patil et al. (2017), Connors et al. (2014) focused on the ability of the methods to recover the probabilities for individual observations in an MNP setting

rather than the underlying choice process parameters at the end of the choice model estimation. They tested four different numbers of alternatives (5, 7, 9, and 15 alternatives) as well as a range of utility values and correlation structures. Their results indicated that, for estimating the probabilities for individual observations, an optimally ordered version of the ME method (to be discussed later in this paper) does much better than the SSJ method with even as many as ten permutations of the abscissae. In addition, they found that the ME method is an order faster than the typical GHK approach in computing the choice probabilities while providing at least the same level of accuracy. A few earlier studies have also done a relatively limited comparison, including Kamakura (1989) who, using a three-to-five alternative set-up in an MNP model, evaluated the ME method with Clark's (1961) approximation and another method proposed by Langdon (1984). He found that the ME method works best relative to the other two, in both evaluating the MVNCD function (as reflected in individual choice probabilities) as well as the underlying MNP model parameters. Joe (1995) tested the SSJ approximation of two different orders (a first order one that entails the evaluation of univariate and bivariate cumulative normal distributions and a more accurate second order one that entails the evaluation of trivariate and quadrivariate cumulative normal distributions) in the context of MVNCD evaluations (rather than MNP parameter recovery). He observed that the SSJ approximation of the first order does better than the ME as well as simulated versions for up to 20 dimensions, though his SSJ approximation is based on averaging over the results of up to 2000 permutations of the abscissae (or all permutations of abscissae if this is less than 2000) for each MVNCD evaluation.

In the current paper, we first propose a streamlined and matrix-based version of the ME method that relies on a single-sided truncation of a multivariate normal distribution in which some variables are truncated while others are not. A number of recent papers have focused on such multivariate distributions and studied the properties of the resulting distributions (see, for example, Kim and Kim, 2015), making use of results related to the moments of truncated multivariate normal distributions (Manjunath and Wilhelm, 2012 and Kan and Robotti, 2017) and using a regression technique (see Kotz et al., 2000, page 70) to obtain the mean and covariance matrices of the untruncated variables from the moments of the truncated variables. We use this approach, except propose a new way to implement this approach using an LDLT decomposition method for the correlation matrix followed by rank 1 or rank 2 updates of the LDLT factorization. This implementation is easier to code and more computationally efficient than the recursive scalar

computations in all earlier implementations of the ME. Using our new matrix-based implementation for the ME algorithm, we also write the analytic matrix-based gradients of the approximated MVNCD function with respect to the input (to the MVNCD function) abscissae and correlation parameters. We also show how, based on our matrix ME implementation, the Trinh and Genz or TG (2015) implementation of the ME method (labeled as the TGME approach) is a substantial simplification that will not provide as accurate MVNCD evaluations as the ME method.[1]

In addition to proposing a streamlined and matrix-based approach for the extant ME method, we propose four new methods for approximating the MVNCD function. The first of these (which we will refer to as the one variate univariate screening or OVUS method) is based on recognizing that, when two untruncated variables are normally distributed, the marginal distribution of one of the untruncated variates given that the other variable is truncated (or screened) is skew-normally distributed and not normally distributed (see Arnold et al., 1993; the ME method, on the other hand, approximates this skew-normal distribution by a normal distribution). The second method (which we will label as the one variate bivariate screening or OVBS method) extends the OVUS method, and uses three variates at a time. The marginal distribution of one variate, given the other two variables are truncated (or screened), takes the OVBS distribution. The third method (which we will label as the bivariate ME or BME method) is based on starting with a quadrivariate normal distribution, using a bivariate truncation scheme for the first two variables, and assuming that the marginal distribution of the third and fourth untruncated variables, given the first two are truncated, remains bivariate normal. Conceptually, this is the extension of the ME method, which is based on univariate conditioning, to a bivariate conditioning mechanism. Trinh and Genz (2015) recently have also proposed a bivariate conditioning generalization of the ME method (algorithms 3.2 and 3.3 of their paper). Again, we will show how the TG version of this bivariate conditioning algorithm (referred to as the TGBME method) is not as accurate as our proposed bivariate conditioning approach. The fourth method (which we will label as the two-variate bivariate screening or TVBS method) combines the

[1] Intuitively, the TG version of the ME method (see their algorithms 2.1 and 2.2) ignores the fact that the variance of a second untruncated marginal element conditional on a first truncated element has a different variance than the variance of the second untruncated element conditional on the first untruncated element. Thus, assume two bivariate normal random variables W_1 and W_2. Then, the marginal distribution of $W_2 \mid W_1 < w_1$ has a different variance than $W_2 \mid W_1$. By ignoring this issue in their algorithms 2.1 and 2.2, TG's ME implementation is not as accurate as the actual ME method (we will demonstrate this through simulation experiments).

bivariate truncation scheme of the second method with the recognition that the bivariate marginal of the untruncated variables in a quadrivariate system is bivariate skewed and not bivariate normal.

The analytic gradients for all these methods have also been coded, tested, and verified. While our way of writing the gradients requires familiarity with matrix differentiation, the matrix-based implementation makes the coding streamlined and compact. Having the analytic gradients of the many MVNCD approximations allows the use of these analytic approximation methods in econometric model estimation to speed up convergence (on the other hand, relying on numerical gradients can be unstable, can lead to convergence problems, and takes an order of magnitude more time, especially as the number of alternatives in the MNP model increases). A complete library of GAUSS matrix programming codes for the proposed matrix-based implementation of the ME method, and for all the new proposed analytic methods as well as the SSJ method (and gradients of all these analytic approaches), are available at http://www.caee.utexas.edu/prof/bhat/LDLT.html.

We evaluate the four proposed analytic MVNCD approximations, as well as the simplified TGME and TGBME methods, with the SSJ and ME methods. For the SSJ method, we consider the first-order approximation that entails the evaluation of only univariate and bivariate cumulative normal distribution functions. We consider both the case of a single permutation (SSJ1), as well as ten random permutations of the abscissae (SSJ10), to compute the MVNCD function (Connors et al., 2014 found little to no benefit of having more than ten permutations in the SSJ method, especially relative to the added time needed). For reference, we also examine the performance of the GHK-Halton simulation method for MVNCD evaluation. In the GHK-Halton, we use 500 Halton draws for drawing from the truncated standard normal distribution, as discussed in detail in Bhat et al. (2010). In the comparison of all the methods, we use different numbers of dimensions of integration (h = 5, 7, 10, 12, 15, 18, and 20). In doing so, we observed a rapid deterioration of the GHK-Halton with 500 Halton draws as the dimensionality increased, and so also added a GHK Halton procedure with 10,000 Halton draws.

In addition to the ability to accurately compute individual MVNCD functions (or, equivalently individual observation MNP choice probabilities), we evaluate performance based on the ability of the methods to recover underlying MNP parameters. Additionally, we ensure that we cover the range of choice probabilities in our evaluation.

The paper is structured as follows. The next section presents the proposed methods. Section 3 presents the evaluation design set-up, clearly identifying the methods being tested and the performance metrics used. Section 4 presents the results. Section 5 concludes the paper by highlighting important findings.

2. THE MVNCD ANALYTIC APPROXIMATION ALGORITHMS

In this section, we first start with two important properties of truncated multivariate normal distributions, which are used in the matrix-oriented implementation of the ME method (that utilizes exact moments for the untruncated variables in a multivariate system, given that a single variable is truncated) as well as all other methods advanced in this paper. We then proceed to a discussion of the ME method and the four new methods proposed in this paper.

2.1. Truncated Multivariate Normal Distributions

We make use of two important properties of truncated multivariate normal distributions, the first dealing with the expected values and variances of single-sided truncations (from above) of univariate and bivariate normally distributed random variables, and the second dealing with the marginal distribution of the vector of untruncated variables when some variables are truncated.

Property 1

Consider a bivariate normally distributed couplet X_1 and X_2:

$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \text{BVN}\left(\begin{matrix} \mu_1 \\ \mu_2 \end{matrix} ; \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix} \right) \sim \text{BVN}(\boldsymbol{\mu}; \boldsymbol{\Sigma}) \tag{1}$$

Let $\mathbf{W}$ be the corresponding standardized equivalent of $\mathbf{X}$; $\mathbf{W} = \boldsymbol{\Gamma}_{\boldsymbol{\Sigma}}^{-1}(\mathbf{X} - \boldsymbol{\mu})$; where $\boldsymbol{\Gamma}_{\boldsymbol{\Sigma}}$ is a diagonal matrix holding the square root of the diagonals of $\boldsymbol{\Sigma}$. Then,

$$\mathbf{W} = \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} \sim \text{BVN}\left(\begin{matrix} 0 \\ 0 \end{matrix} ; \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \right) \tag{2}$$

The expected value vector of $Y_1 = X_1 \,|\, (X_1 < x_1, X_2 < x_2)$ and $Y_2 = X_2 \,|\, (X_1 < x_1, X_2 < x_2)$ is as follows:

$$E(\mathbf{Y}) = E\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} \widetilde{\mu}_1 \\ \widetilde{\mu}_2 \end{pmatrix} = \widetilde{\boldsymbol{\mu}} = \begin{pmatrix} \mu_1 + \sigma_1 \lambda_1 \\ \mu_2 + \sigma_2 \lambda_2 \end{pmatrix}, \text{ where} \tag{3}$$

$$\lambda_i = E(Z_i) = E\left(W_i \mid (W_1 < w_1, W_2 < w_2)\right) = -\left(\frac{\delta_i + \rho\delta_j}{\Phi_2(w_1, w_2, \rho)}\right), \text{ where } Z_i = W_i \mid (W_1 < w_1, W_2 < w_2),$$
$$\delta_i = \phi(w_i)\Phi\left(\frac{w_j - \rho w_i}{\sqrt{1-\rho^2}}\right), \; w_i = \frac{x_i - \mu_i}{\sigma_i}, \; i = 1,2; j \neq i, j = 1,2. \tag{4}$$

$\Phi_2(.)$ above is the bivariate standard normal cumulative distribution function, $\Phi(.)$ is the univariate standard normal cumulative distribution function, and $\phi(.)$ is the univariate standard normal density function. The covariance matrix of $\mathbf{Y}$ is:

$$Cov(\mathbf{Y}) = \mathbf{\Omega} = \begin{bmatrix} \ddot{\sigma}_1^2 & \ddot{\sigma}_{12} \\ \ddot{\sigma}_{12} & \ddot{\sigma}_2^2 \end{bmatrix} = \mathbf{\Gamma}_{\Sigma}\, Cov(\mathbf{Z})\, \mathbf{\Gamma}_{\Sigma}, \text{ where } \mathbf{Z} = (Z_1, Z_2)', \; Cov(Z) = \begin{bmatrix} \tilde{\sigma}_1^2 & \tilde{\sigma}_{12} \\ \tilde{\sigma}_{12} & \tilde{\sigma}_2^2 \end{bmatrix}, \text{ with} \tag{5}$$

$$\tilde{\sigma}_i^2 = 1 - \frac{1}{\Phi_2(w_1, w_2, \rho)}\left\{w_i\delta_i + \rho^2 w_j\delta_j - (1-\rho^2)\,\rho\,\phi_2(w_1, w_2, \rho)\right\} - \lambda_i^2, \; i = 1,2; j \neq i, j = 1,2$$

$$\tilde{\sigma}_{12} = \rho - \frac{1}{\Phi_2(w_1, w_2, \rho)}\left\{\rho w_1\delta_1 + \rho w_2\delta_2 - (1-\rho^2)\,\phi_2(w_1, w_2, \rho)\right\} - \lambda_1\lambda_2$$

$\phi_2(.)$ above represent the bivariate standard normal probability density function. The proof is in Appendix A.

When considering only a single normal variable that is truncated (say X_1), then, $x_2 = w_2 = \infty, \; \delta_2 = 0, \; \delta_1 = \phi(w_1), \; \Phi_2(w_1, w_2, \rho) = \Phi(w_1), \; \phi_2(w_1, w_2, \rho) = 0, \lambda_1 = -\frac{\phi(w_1)}{\Phi(w_1)}$, and the formulas above for Y_1 collapse to those of the single truncated univariate case:

$$E(Y_1) = \mu_1 + \sigma_1\lambda_1, \; Var(Y_1) = \sigma_1^2\,(1 + \lambda_1 w_1 - \lambda_1^2). \tag{6}$$

Property 2

Consider a *H*-dimensional vector $\mathbf{X}$ of multivariate normally distributed variables $\mathbf{X} = (X_1, X_2, ...X_H)$; $\mathbf{X} \sim MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Let the first *G* variables (*G*<*H*) be truncated, while the remaining are not truncated. In our notation above,

$$\vec{\mathbf{Y}}_1 = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_G \end{pmatrix} = \begin{pmatrix} X_1 \mid X_1 < x_1, X_2 < x_2, ..., X_G < x_G \\ X_2 \mid X_1 < x_1, X_2 < x_2, ..., X_G < x_G \\ \vdots \\ X_G \mid X_1 < x_1, X_2 < x_2, ..., X_G < x_G \end{pmatrix}. \tag{7}$$

Let $E(\tilde{\mathbf{Y}}_1) = \tilde{\boldsymbol{\mu}}_1$ and $Cov(\tilde{\mathbf{Y}}_1) = \boldsymbol{\Omega}_1$. Partition the $\mathbf{X}$ vector into two sub-vectors: $\mathbf{X} = (\tilde{\mathbf{X}}_1', \tilde{\mathbf{X}}_2')'$, with $\tilde{\mathbf{X}}_1 = (X_1, X_2, ...X_G)'$ and $\tilde{\mathbf{X}}_2 = (X_{G+1}, X_{G+2}, ...X_H)'$. Correspondingly partition the mean and covariance matrices of $\mathbf{X}$ as follows: $\boldsymbol{\mu} = (\boldsymbol{\mu}_1', \boldsymbol{\mu}_2')'$ and $\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\psi}_{11} & \boldsymbol{\psi}_{21}' \\ \boldsymbol{\psi}_{21} & \boldsymbol{\psi}_{22} \end{bmatrix}$. Let $\tilde{\mathbf{Y}}_2 = \tilde{\mathbf{X}}_2 \,|\, (\tilde{\mathbf{X}}_1 < \tilde{\mathbf{x}}_1)$, where $\tilde{\mathbf{x}}_1 = (x_1, x_2, ..., x_G)$. Then, we can write:

$$E\begin{pmatrix} \tilde{\mathbf{Y}}_1 \\ \tilde{\mathbf{Y}}_2 \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{\mu}}_1 \\ \boldsymbol{\mu}_2 + \boldsymbol{\psi}_{21}\boldsymbol{\psi}_{11}^{-1}(\tilde{\boldsymbol{\mu}}_1 - \boldsymbol{\mu}_1) \end{pmatrix}; \; Cov\begin{pmatrix} \tilde{\mathbf{Y}}_1 \\ \tilde{\mathbf{Y}}_2 \end{pmatrix} = \begin{bmatrix} \boldsymbol{\Omega}_1 & \boldsymbol{\Omega}_1 \boldsymbol{\psi}_{11}^{-1}\boldsymbol{\psi}_{21}' \\ \boldsymbol{\psi}_{21}\boldsymbol{\psi}_{11}^{-1}\boldsymbol{\Omega}_1 & \boldsymbol{\psi}_{22} - \boldsymbol{\psi}_{21}(\boldsymbol{\psi}_{11}^{-1} - \boldsymbol{\psi}_{11}^{-1}\boldsymbol{\Omega}_1\boldsymbol{\psi}_{11}^{-1})\boldsymbol{\psi}_{21}' \end{bmatrix} \quad (8)$$

This property is based on Pearson (1903) and Aitken (1934), and is also presented in Kotz et al. (2000) and Manjunath and Wilhelm (2012). It is used by Kim and Kim (2015) in obtaining the first two moments of their general class of rectangle-screened multivariate normal distributions. The ME method (discussed in the next section in more detail) is based on first applying Equation (8) to the truncation of X_1 (such that $X_1 < x_1$), and determining the expected value vector and covariance matrix of the vector $\tilde{\mathbf{Y}}_2 = (Y_2, Y_3, ..., Y_H)$. So far, all is fine. Then, it starts with the vector $\tilde{\mathbf{Y}}_2 = (Y_2, Y_3, ..., Y_H)$, assumes this to be multivariate normal, applies a truncation next on X_2 (such that $X_2 < x_2$), and applies Equation (8) again to get the expected value vector and covariance matrix of the resulting new $\tilde{\mathbf{Y}}_3 = (Y_3, Y_4, ..., Y_H)$. This process is continued. However, after the first truncation on X_1, $\tilde{\mathbf{Y}}_2$ is not multivariate normal. Thus the use of Equation (8) is not strictly correct, but is used as an approximation to obtain the expected value vector and covariance matrix of $\tilde{\mathbf{Y}}_3 = (Y_3, Y_4, ..., Y_H)$. To be precise, $\tilde{\mathbf{Y}}_2$ takes a multivariate skew distribution form (see Lee and McLachlan, 2013; though the truncation of X_1 is from above and not below as in the standard skew distribution form).

2.2. The ME Method

The ME method is based on univariate conditioning. The fundamental concept behind the univariate conditioning mechanism, used also in the GHK procedure as well as the ME, is that the MVNCD function may be written conveniently as a sequence of UCCNCDs. More specifically, let $(W_1, W_2, W_3, ..., W_H)$ be a multivariate normally distributed random vector with zero means,

variances of 1, and a correlation matrix $\boldsymbol{\Sigma}$ ($H > 2$). Then, interest centers on approximating the following orthant probability:

$$\Phi_H(w_1, w_2, ..., w_H : \boldsymbol{\Sigma}) = \Pr(\boldsymbol{W} < \boldsymbol{w}) = \Pr(W_1 < w_1,\ W_2 < w_2,\ W_3 < w_3, ...,\ W_H < w_H). \quad (9)$$

The above joint probability may be written as the product of a univariate marginal probability and univariate conditional probabilities as follows ($H \geq 2$):

$$\Pr(\boldsymbol{W} < \boldsymbol{w}) = \Pr(W_1 < w_1) \times \prod_{h=1}^{H-1} \Pr(W_{h+1} < w_{h+1} \mid W_1 < w_1,\ W_2 < w_2,\ W_3 < w_3, ...,\ W_h < w_h). \quad (10)$$

Earlier discussions and implementations of the ME method (see Kamakura, 1989, Rice et al., 1979, and Hutmacher and French, 2011) use rather notationally intensive recursive scalar manipulations based on Property 2. Our implementation, on the other hand, is based directly off the applications of the two properties discussed in the earlier section, supplemented with a much more streamlined way of implementing Property 2 that obtains the results in Equation (8) in a quick and simple matrix-based form. To explicate our new proposal for ME implementation, we first state the following two additional properties.

Property 3

Consider a H-dimensional vector $\mathbf{X}$ of multivariate normally distributed variables $\mathbf{X} = (X_1, X_2, ..., X_H)$; $\mathbf{X} \sim MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Let $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{D}\mathbf{L}'$ be a G-block decomposition, where $\mathbf{L}$ is the unit lower-triangular matrix and $\mathbf{D}$ is a block diagonal matrix with each block diagonal of size G (this is usually referred to as the LDLT decomposition of the covariance matrix $\boldsymbol{\Sigma}$). Consider the notation in Property 2 with the number of truncated variables being G. Based on the LDLT decomposition, we can write:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\psi}_{11} & \boldsymbol{\psi}_{21}' \\ \boldsymbol{\psi}_{21} & \boldsymbol{\psi}_{22} \end{bmatrix} = \mathbf{L}\mathbf{D}\mathbf{L}' = \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 & \mathbf{0}' \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_1 & \mathbf{L}_{21}' \\ \mathbf{0} & \mathbf{L}_{22}' \end{bmatrix} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{D}_1\mathbf{L}_{21}' \\ \mathbf{L}_{21}\mathbf{D}_1 & \mathbf{L}_{21}\mathbf{D}_1\mathbf{L}_{21}' + \mathbf{L}_{22}\mathbf{D}_2\mathbf{L}_{22}' \end{bmatrix} \quad (11)$$

Then, Equation (8) may be equivalently written as:

$$E\begin{pmatrix} \bar{\mathbf{Y}}_1 \\ \tilde{\mathbf{Y}}_2 \end{pmatrix} = \boldsymbol{\mu} + \mathbf{L}\begin{pmatrix} \tilde{\boldsymbol{\mu}}_1 - \boldsymbol{\mu}_1 \\ \mathbf{0} \end{pmatrix}, \quad (12)$$

$$Cov\begin{pmatrix} \bar{\mathbf{Y}}_1 \\ \tilde{\mathbf{Y}}_2 \end{pmatrix} = \mathbf{L}\begin{bmatrix} \boldsymbol{\Omega}_1 & \mathbf{0}' \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix}\mathbf{L}' = \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Omega}_1 & \mathbf{0}' \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_1 & \mathbf{L}_{21}' \\ \mathbf{0} & \mathbf{L}_{22}' \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Omega}_1 & \boldsymbol{\Omega}_1\mathbf{L}_{21}' \\ \mathbf{L}_{21}\boldsymbol{\Omega}_1 & \mathbf{L}_{21}\boldsymbol{\Omega}_1\mathbf{L}_{21}' + \mathbf{L}_{22}\mathbf{D}_2\mathbf{L}_{22}' \end{bmatrix},$$

where $\tilde{\boldsymbol{\mu}}_1$ and $\boldsymbol{\Omega}_1$ are the expected value and covariance matrix of the truncated normally

distributed vector $\tilde{\mathbf{Y}}_1$ (in the case that *G*=1 or *G*=2, which are the cases that we will use in the current paper, the expressions for $\tilde{\boldsymbol{\mu}}_1$ and $\boldsymbol{\Omega}_1$ are available from property 1). The proof is straightforward based on substituting for $\boldsymbol{\psi}_{11}$, $\boldsymbol{\psi}_{21}$, and $\boldsymbol{\psi}_{22}$ in Equation (8) by the right side of Equation (11). The above property obviates the need to undertake the relatively more expensive manipulations of Equation (8). The advantage of the method will become more clear when combined with Property 4 below.

Property 4

In the ME algorithm, there is a need to sequentially truncate on successive variables at the univariate level. Let the first variable be truncated, while the remaining are not truncated. In our notation earlier, $Y_1 = X_1 \mid X_1 < x_1$, and, correspondingly, $Z_1 = W_1 \mid W_1 < w_1$ with $w_1 = \dfrac{x_1 - \mu_1}{\sigma_1}$. Also, $\boldsymbol{\mu} = (\mu_1, \boldsymbol{\mu}_2')'$. Then, from the univariate version of Property 1, we have $E(Y_1) = \tilde{\mu}_1 = \mu_1 + \sigma_1\lambda_1$, $Var(Y_1) = \Omega_1 = \sigma_1^2(1 + \lambda_1 w_1 - \lambda_1^2) = \sigma_1^2\theta_1^2$, where $E(Z_1) = \lambda_1$ and $Var(Z_1) = \theta_1^2 = (1 + \lambda_1 w_1 - \lambda_1^2)$. Using Property 3, the mean vector and covariance matrix of $(Y_1, \tilde{\mathbf{Y}}_2')'$ can be expressed as follows:

$$E\begin{pmatrix} Y_1 \\ \tilde{\mathbf{Y}}_2 \end{pmatrix} = \boldsymbol{\mu} + \mathbf{L}\begin{pmatrix} \tilde{\mu}_1 - \mu_1 \\ \mathbf{0} \end{pmatrix};\ Cov\begin{pmatrix} Y_1 \\ \tilde{\mathbf{Y}}_2 \end{pmatrix} = \mathbf{L}\begin{bmatrix} \Omega_1 & \mathbf{0}' \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix}\mathbf{L}', \tag{13}$$

where the matrix sandwiched between the **L** matrices in the last expression above is the **D** matrix except for the first element being replaced by the Ω_1 term (representing the variance of the truncated first element). Thus, in the context of the previous property, $E(\tilde{\mathbf{Y}}_2) = \boldsymbol{\mu}_2 + \mathbf{L}_{21}(\tilde{\mu}_1 - \mu_1)$ and $Cov(\tilde{\mathbf{Y}}_2) = \mathbf{L}_{21}\Omega_1\mathbf{L}_{21}' + \mathbf{L}_{22}\mathbf{D}_2\mathbf{L}_{22}'$, The next step would be to partition $\tilde{\mathbf{X}}_2$ as $\tilde{\mathbf{X}}_2 = (X_2, \tilde{\mathbf{X}}_3)'$, where $\tilde{\mathbf{X}}_3 = (X_3, ..., X_H)'$ and then implement Property 2 again (in an approximate way now, as discussed earlier) by truncating on X_2. When using Property 3 instead of Property 2, the analyst will need to obtain the LDLT decomposition of $Cov(\tilde{\mathbf{Y}}_2) = \mathbf{L}_{21}\Omega_1\mathbf{L}_{21}' + \mathbf{L}_{22}\mathbf{D}_2\mathbf{L}_{22}'$. Herein lies the added advantage of using Property 3 rather than the equivalent Property 2 (to obtain $E(\tilde{\mathbf{Y}}_3)$ and $Cov(\tilde{\mathbf{Y}}_3)$). This is because the LDLT factorization of $Cov(\tilde{\mathbf{Y}}_2) = \mathbf{L}_{21}\Omega_1\mathbf{L}_{21}' + \mathbf{L}_{22}\mathbf{D}_2\mathbf{L}_{22}'$ can be obtained as a simple rank-1 update of the previously obtained LDLT decomposition of $\boldsymbol{\Sigma} = \mathbf{LDL}'$,

because Ω_1 is a scalar (so the first term $\mathbf{L}_{21}\Omega_1\mathbf{L}'_{21}$ is the rank-one update of the already existing LDLT decomposition $\mathbf{L}_{22}\mathbf{D}_2\mathbf{L}'_{22}$). Formally, we may write:

$$\mathbf{L}_{21}\Omega_1\mathbf{L}'_{21} + \mathbf{L}_{22}\mathbf{D}_2\mathbf{L}'_{22} = \mathbf{L}_{new}\mathbf{D}_{new}\mathbf{L}'_{new} \tag{14}$$

Pre- and post-multiplying both sides of the above equation by $\mathbf{L}_{22}^{-1}$ and $\left(\mathbf{L}_{22}^{-1}\right)'$, and re-arranging, we obtain:

$$\mathbf{G}\Omega_1\mathbf{G}' + \mathbf{D}_2 = \mathbf{L}_{22}^{-1}\mathbf{L}_{new}\mathbf{D}_{new}\mathbf{L}'_{new}(\mathbf{L}_{22}^{-1})', \text{ where } \mathbf{G} = \mathbf{L}_{22}^{-1}(\mathbf{L}_{21}) \text{ [a } (H-1)\times 1 \text{ matrix]}. \tag{15}$$

We can very quickly (in order $O(H)$ time) find the $\tilde{\mathbf{L}}\tilde{\mathbf{D}}\tilde{\mathbf{L}}'$ factorization for the left side of the above equation $\mathbf{G}\Omega_1\mathbf{G}' + \mathbf{D}_2$ using a triangular factorization-based updating algorithm by using suitable Givens rotations (see Golub and Van Loan, 2013), and then $\mathbf{L}_{new} = \mathbf{L}_{22}\tilde{\mathbf{L}}$ and $\mathbf{D}_{new} = \tilde{\mathbf{D}}$. Our approach is basically a combination of rank-one updating with a deletion of a row/column of the original covariance matrix $\mathbf{\Sigma} = \mathbf{LDL}'$.

Property 4 may be formally stated as follows. At each iteration (i.e., truncation of a variable) of the ME method, the needed LDLT decomposition to apply Property 3 constitutes a simple update of the original (only once-to-be-computed) LDLT decomposition of $\mathbf{\Sigma} = \mathbf{LDL}'$ (that is, there is no need to repeatedly compute LDLT decompositions, which can be expensive). Specifically, we first obtain **L** and **D** in the first (and only) LDLT decomposition of the original covariance (correlation) matrix **Σ**, next truncate on the variable X_1, compute $\tilde{\mu}_1 = E(Y_1)$ and $\Omega_1 = Var(Y_1)$, and obtain $E(\tilde{\mathbf{Y}}_2)$, then develop the LDLT decomposition of $\tilde{\mathbf{\Sigma}}_2 = Cov(\tilde{\mathbf{Y}}_2)$ as $\mathbf{L}_2^{new}$ and $\mathbf{D}_2^{new}$ employing a simple updating mechanism (that uses only **L**, **D**, and the variance of the truncated X_1 (i.e., $\Omega_1 = Var(Y_1)$) and obtain the first element of $\tilde{\mathbf{\Sigma}}_2$ simply as the first element of $\mathbf{D}_2^{new}$. Next, we truncate on the variable X_2, compute $E(Y_2)$ and $\Omega_2 = Var(Y_2)$, and obtain $E(\tilde{\mathbf{Y}}_3)$, develop the LDLT decomposition of $\tilde{\mathbf{\Sigma}}_3$ to obtain $\mathbf{L}_3^{new}$ and $\mathbf{D}_3^{new}$ using a simple updating procedure, and obtain the first element of $\tilde{\mathbf{\Sigma}}_3 = Cov(\tilde{\mathbf{Y}}_3)$ as the first element of $\mathbf{D}_3^{new}$. This process is continued.

The pseudo-code for the LDLT updating algorithm applicable to a rank-1 update with a single row/column deletion or a rank-2 update with the deletion of two rows/columns is available at http://www.caee.utexas.edu/prof/bhat/ABSTRACTS/LDLT/Pseudo-code.pdf. The rank-2 update will be useful for some of the new approximation procedures proposed later in this paper

(even though only the rank-one update with a single row/column deletion is used in our ME implementation). The update procedure takes much less time (is of $O(H)$ order in time) than explicitly performing the LDLT decomposition (which is an $O(H^3)$ operation) at each of the H–1 iterations of the ME method). All that is needed in our implementation is the original LDLT decomposition, which is then updated fast at each truncation of a variable once the expected value and variance of the previously truncated variable is in place.

We are now able to state our implementation of the ME method in the following pseudo-code (in terms of the quantities already defined; $H \geq 2$):

(1) Compute $P_1 = \Phi(w_1)$

(2) Compute the LDLT decomposition $\mathbf{LDL}'$ of the original covariance matrix $\mathbf{\Sigma}$. For notational convention purposes, set $\boldsymbol{\pi}_1 = E(\widetilde{\mathbf{Y}}_1) = E(\mathbf{X}) = \boldsymbol{\mu} = \mathbf{0}$, and $\mathbf{L}_1^{new} = \mathbf{L}, \mathbf{D}_1^{new} = \mathbf{D}$.

(3) For h = 1 to H–1;

(4) Truncate on X_h (that is, $X_h < x_h$) and obtain $E(Y_h) = \widetilde{\mu}_h$ and $Var(Y_h) = \Omega_h$ using Equation (6) in Property 1, with μ_h (in the computation of $\widetilde{\mu}_h$) being replaced by the first element of $\boldsymbol{\pi}_h = E(\widetilde{\mathbf{Y}}_h)$ and σ_h obtained as the square root of the first element of $\mathbf{D}_h^{new}$ $(= \mathbf{D}_h^{new}[1,1])$, and compute $\boldsymbol{\pi}_{h+1} = E(\widetilde{\mathbf{Y}}_{h+1}) = \boldsymbol{\pi}_h[2 : H - h + 1] + \mathbf{L}_h^{new}[2 : H - h + 1,1]\ (\widetilde{\mu}_h - \boldsymbol{\pi}_h[1])$ using Property 3.

(5) Implement an LDLT decomposition of $\widetilde{\mathbf{\Sigma}}_{h+1} = Cov(\widetilde{\mathbf{Y}}_{h+1})$ using a call to the LDLT rank-1 updating algorithm with inputs $\{\mathbf{L}_h^{new}, \mathbf{D}_h^{new}, \Omega_h\}$ to obtain $\mathbf{L}_{h+1}^{new}$ and $\mathbf{D}_{h+1}^{new}$.

(6) Obtain $E(Y_{h+1})$ as the first entry of the vector $\boldsymbol{\pi}_{h+1} = E(\widetilde{\mathbf{Y}}_{h+1})$ and $Var(Y_{h+1})$ as the first entry of the matrix $\mathbf{D}_{h+1}^{new}$. Compute $P_{h+1} = \Phi\left(\dfrac{w_{h+1} - E(Y_{h+1})}{\sqrt{Var(Y_{h+1})}}\right)$.

(7) End for

(8) Return $(\prod_{h=1}^{H} P_h)$.

It is important to note that Trinh and Genz's (TG's) (2015) implementation in their algorithms 2.1 and 2.2 do not represent the ME method. When taken in the context of the ME implementation proposed in this paper, the TG approach essentially ignores the fact that the variance of a sub-

vector ($\widetilde{\mathbf{Y}}_h$ in our notation) conditional on truncation on another sub-vector (when both the untruncated sub-vectors have a multivariate distribution) is not the same as the variance of the untruncated $\widetilde{\mathbf{X}}_h$ vector conditional on the other untruncated sub-vector. Specifically, in Equation (13) of our ME implementation, Trinh and Genz (2015) do not compute the Ω_1 variance term, and simply maintain the same original covariance matrix $\mathbf{\Sigma}$ without any updating after truncation. In other words, there is no computation of Ω_1 in step 4 of our ME implementation, and step 5 of the algorithm above is completely ignored in the TG algorithms (with **D** and **L** trivially revised at each *h*-step as pure submatrices of the original **D** and **L** matrices -- $\mathbf{D}_{h+1}^{new} = \mathbf{D}[h+1:H-h+1, h+1:H-h+1]$, $\mathbf{L}_{h+1}^{new} = \mathbf{L}[h+1:H-h+1, h+1:H-h+1]$). The result is that TG's approach overestimates the MVNCD approximation relative to the ME method, with larger errors relative to the ME method. As an illustration, consider a simple bivariate normal cumulative distribution function evaluation of W_1 and W_2, with $\boldsymbol{\mu} = (0,0)'$ and $\mathbf{\Sigma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix}$. The intent is to approximate $P(W_1 < 0.3,\, W_2 < 1)$. In our ME implementation, first $P_1 = \Phi(0.3) = 0.61791$. The LDLT decomposition of $\mathbf{\Sigma}$ provides $\mathbf{L}_1 = \begin{bmatrix} 1 & 0 \\ 0.4 & 1 \end{bmatrix}$ and $D_1 = \begin{bmatrix} 1 & 0 \\ 0.0 & 0.84 \end{bmatrix}$. Truncating on $W_1 < 0.3$ provides $\widetilde{\mu}_1 = -0.61722$ and $\Omega_1 = 0.43387$ (expressed to the closest five digits of accuracy for presentation ease), and $\boldsymbol{\pi}_2 = E(\widetilde{\mathbf{Y}}_2) = 0 + 0.4\,(-0.61722 - 0) = -0.24689$. Implementing step 5 and step 6 provides $\mathbf{L}_2 = 1.0$, $\mathbf{D}_2 = 0.90942\,(= 1 - \rho^2 + \rho^2\Omega_1)$, and $Var(Y_2) = 0.90942$, and $P_2 = \Phi\left(\frac{1-(-0.24689)}{\sqrt{0.90942}}\right) = 0.90448043$, and $P(W_1 < 0.3,\, W_2 < 1) \approx 0.61791 * 0.90448 = 0.55889$.

In the TG algorithms, $P_1 = \Phi(0.3) = 0.61791$, as earlier. But Ω_1 is not computed, $\boldsymbol{\pi}_2 = E(\widetilde{\mathbf{Y}}_2) = 0 + 0.4\,(-0.61722 - 0) = -0.24689$ (as earlier), step 5 is not implemented, $Var(Y_2)$ is assigned the $\mathbf{D}_1[2,2]$ value of 0.84 $(= 1 - \rho^2)$, and $P_2 = \Phi\left(\frac{1-(-0.24689)}{\sqrt{0.84}}\right) = 0.91316$, and $P(W_1 < 0.3,\, W_2 < 1) \approx 0.61791 * 0.91316 = 0.56425$. The true value of $P(W_1 < 0.3,\, W_2 < 1)$ is 0.55915. Clearly, our ME implementation is closer to the true value, while the TG implementation

is an overestimate. In fact, this problem permeates into the bivariate conditioning algorithm too proposed in TG.

2.3. New Methods

In this paper, we propose four new approaches to approximate the MVNCD function. Two of these methods are based on univariate truncation (that is, at each step of the approach, a single variable is truncated; in the context of property 2 earlier, G=1), and two others are based on bivariate truncation (that is, at each step of the approach, two variables are simultaneously truncated; in the context of property 2 earlier, G=2). In the following presentation, the same notation is used as earlier.

2.3.1. Univariate Truncation Approaches

2.3.1.1 The One-Variate Univariate Screening (OVUS) Method

The ME method, as already discussed, is based on successive truncations on X_h (that is, $X_h < x_h$), and the assumption at each step that the first element of $\widetilde{\mathbf{Y}}_{h+1}$ is normally distributed. However, this method can be improved by noting that, if we assume that Y_{h+1} and Y_{h+2} are bivariate normally distributed at each step, the distribution of Y_{h+2} given Y_{h+1} is a type of a skew-normal distribution and does not have a normal distribution as assumed in the ME method. Using the terminology of Kim and Kim (2015), Y_{h+2} is a one-variate univariate screened variable (that is, is based on selection or screening on one variable Y_{h+1}). This is a simple, yet significant improvement over the ME method. Further, given the speed at which a bivariate normal cumulative distribution function can be computed today, the computation time is not likely to be substantially more than the ME implementation, as we will note later. Formally, the algorithm is as follows ($H \geq 2$):

(1) Compute $P_1 = \Phi(w_1) \times \dfrac{\Phi_2(w_1, w_2, \rho_{12})}{\Phi(w_1)} = \Phi_2(w_1, w_2, \rho_{12})$. If H=2, STOP.

(2) Compute the LDLT decomposition $\mathbf{LDL}'$ of the original covariance matrix $\mathbf{\Sigma}$. For notational convention purposes, set $\boldsymbol{\pi}_1 = E(\widetilde{\mathbf{Y}}_1) = E(\mathbf{X}) = \boldsymbol{\mu} = \mathbf{0}$, and $\mathbf{L}_1^{new} = \mathbf{L}, \mathbf{D}_1^{new} = \mathbf{D}$.

(3) For h = 1 to H–2;

(4) Truncate on X_h (that is, $X_h < x_h$) and obtain $E(Y_h) = \widetilde{\mu}_h$ and $Var(Y_h) = \Omega_h$ using Equation (6) in Property 1, with μ_h (in the computation of $\widetilde{\mu}_h$) being replaced by the first element of $\boldsymbol{\pi}_h = E(\widetilde{\mathbf{Y}}_h)$ and σ_h obtained as the square root of the first element of $\mathbf{D}_h^{new}$ $(= \mathbf{D}_h^{new}[1,1])$, and compute $\boldsymbol{\pi}_{h+1} = E(\widetilde{\mathbf{Y}}_{h+1}) = \boldsymbol{\pi}_h[2:H-h+1] + \mathbf{L}_h^{new}[2:H-h+1,1]\,(\widetilde{\mu}_h - \boldsymbol{\pi}_h[1])$ using Property 3.

(5) Implement an LDLT decomposition of $\tilde{\boldsymbol{\Sigma}}_{h+1} = Cov(\tilde{\mathbf{Y}}_{h+1})$ using a call to the LDLT rank-1 updating algorithm with inputs $\{\mathbf{L}_h^{new}, \mathbf{D}_h^{new}, \Omega_h\}$ to obtain $\mathbf{L}_{h+1}^{new}$ and $\mathbf{D}_{h+1}^{new}$.

(6) Obtain $\widetilde{\boldsymbol{\lambda}}_{h+1} = (E(Y_{h+1}), E(Y_{h+2}))'$ as the first two entries of the vector $\boldsymbol{\pi}_{h+1} = E(\widetilde{\mathbf{Y}}_{h+1})$, $\widetilde{\mathbf{w}}_{h+1} = (w_{h+1}, w_{h+2})'$, and the covariance matrix $\boldsymbol{\Xi}_{h+1}$ of Y_{h+1} and Y_{h+2} as $\left(\mathbf{L}_{h+1}^{new}[1:2,1:2]\right)\mathbf{D}_{h+1}^{new}[1:2,1:2]\left(\mathbf{L}_{h+1}^{new}[1:2,1:2]\right)'$. $\boldsymbol{\Xi}_{h+1}$ is basically the covariance sub-matrix of the first two rows and first two columns of $\tilde{\boldsymbol{\Sigma}}_{h+1}$.[2] Let $\widetilde{\boldsymbol{\Xi}}_{h+1} = \boldsymbol{\Gamma}_{h+1}^{-1}\boldsymbol{\Xi}_{h+1}\boldsymbol{\Gamma}_{h+1}^{-1}$, where $\boldsymbol{\Gamma}_{h+1}$ is a diagonal matrix of the square root of the diagonal elements of $\boldsymbol{\Xi}_{h+1}$. Compute

$$P_{h+1} = \frac{\Phi_2(\boldsymbol{\Gamma}_{h+1}^{-1}(\widetilde{\mathbf{w}}_{h+1} - \widetilde{\boldsymbol{\lambda}}_{h+1}); \widetilde{\boldsymbol{\Xi}}_{h+1})}{\Phi\left[(w_{h+1} - E(Y_{h+1}) / \sqrt{Var(Y_{h+1})}\right]}.$$

(7) End for

(8) Return ($P_1 \times \prod_{h=1}^{H-2} P_{h+1}$).

2.3.1.2. The One-Variate Bivariate Screening (OVBS) Method

The univariate screening method can be extended to a bivariate screening method. Specifically, if we assume that Y_{h+1}, Y_{h+2}, and Y_{h+3} are trivariate normally distributed at each step, Y_{h+3} given Y_{h+1} and Y_{h+2} is a one-variate variable based on bivariate screening. Using this approach to approximate the MVNCD function should improve accuracy, though it also will increase the computation time marginally, because it entails the evaluation of trivariate normal cumulative distribution functions. Formally, the algorithm is as follows ($H > 3$):

[2] Because of the lower-triangular nature of the **L** matrix and the diagonal nature of the **D** matrix, $\boldsymbol{\Xi}_{h+1}$ can be computed in a straight forward manner as $\left(\mathbf{L}_{h+1}^{new}[1:2,1:2]\right)\mathbf{D}_{h+1}^{new}[1:2,1:2]\left(\mathbf{L}_{h+1}^{new}[1:2,1:2]\right)'$ instead of computing the entire $\tilde{\boldsymbol{\Sigma}}_{h+1} = Cov(\tilde{\mathbf{Y}}_{h+1})$ matrix and then taking the sub-matrix of this matrix.

(1) Compute $P_1 = \Phi(w_1) \times \frac{\Phi_2(w_1, w_2 : \tilde{\boldsymbol{\Xi}}_1)}{\Phi(w_1)} \times \frac{\Phi_3(w_1, w_2, w_3; \tilde{\boldsymbol{\Lambda}}_1)}{\Phi_2(w_1, w_2; \tilde{\boldsymbol{\Xi}}_1)} = \Phi_3(w_1, w_2, w_3; \boldsymbol{\Lambda}_1)$, where $\tilde{\boldsymbol{\Lambda}}_1$ is the marginal trivariate correlation matrix of the first three rows and first three columns of $\boldsymbol{\Sigma}$ and $\tilde{\boldsymbol{\Xi}}_1$ is the marginal bivariate correlation matrix of the first two rows and two columns of $\boldsymbol{\Sigma}$. If H=3, STOP.

(2) Compute the LDLT decomposition $\mathbf{LDL}'$ of the original covariance matrix $\boldsymbol{\Sigma}$. For notational convention purposes, set $\boldsymbol{\pi}_1 = E(\tilde{\mathbf{Y}}_1) = E(\mathbf{X}) = \boldsymbol{\mu} = \mathbf{0}$, and $\mathbf{L}_1^{new} = \mathbf{L}, \mathbf{D}_1^{new} = \mathbf{D}$.

(3) For h = 1 to H–3;

(4) Truncate on X_h (that is, $X_h < x_h$) and obtain $E(Y_h) = \tilde{\mu}_h$ and $Var(Y_h) = \Omega_h$ using Equation (6) in Property 1, with μ_h (in the computation of $\tilde{\mu}_h$) being replaced by the first element of $\boldsymbol{\pi}_h = E(\tilde{\mathbf{Y}}_h)$ and σ_h obtained as the square root of the first element of $\mathbf{D}_h^{new}$ $(= \mathbf{D}_h^{new}[1,1])$, and compute $\boldsymbol{\pi}_{h+1} = E(\tilde{\mathbf{Y}}_{h+1}) = \boldsymbol{\pi}_h[2 : H-h+1] + \mathbf{L}_h^{new}[2 : H-h+1,1]\ (\tilde{\mu}_h - \boldsymbol{\pi}_h[1])$ using Property 3.

(5) Implement an LDLT decomposition of $\tilde{\boldsymbol{\Sigma}}_{h+1} = Cov(\tilde{\mathbf{Y}}_{h+1})$ using a call to the LDLT rank-1 updating algorithm with inputs $\{\mathbf{L}_h^{new}, \mathbf{D}_h^{new}, \Omega_h\}$ to obtain $\mathbf{L}_{h+1}^{new}$ and $\mathbf{D}_{h+1}^{new}$.

(6) Obtain $\boldsymbol{\lambda}_{h+1} = (E(Y_{h+1}), E(Y_{h+2}))'$ as the first two entries of the vector $\boldsymbol{\pi}_{h+1} = E(\tilde{\mathbf{Y}}_{h+1})$, and $\boldsymbol{\theta}_{h+1} = (E(Y_{h+1}), E(Y_{h+2}), E(Y_{h+3}))'$ as the first three entries of the vector $\boldsymbol{\pi}_{h+1} = E(\tilde{\mathbf{Y}}_{h+1})$. Let $\tilde{\mathbf{w}}_{h+1} = (w_{h+1}, w_{h+2})'$ and $\vec{\mathbf{w}}_{h+1} = (w_{h+1}, w_{h+2}, w_{h+3})'$. Let $\boldsymbol{\Xi}_{h+1}$ be the covariance sub-matrix of the first two rows and two columns of $\tilde{\boldsymbol{\Sigma}}_{h+1}$, and $\boldsymbol{\Lambda}_{h+1}$ be the covariance sub-matrix obtained as the first three rows and three columns of $Cov(\tilde{\mathbf{Y}}_{h+1})$.[3] Let $\tilde{\boldsymbol{\Xi}}_{h+1} = \boldsymbol{\Gamma}_{h+1}^{-1} \boldsymbol{\Xi}_{h+1} \boldsymbol{\Gamma}_{h+1}^{-1}$, where $\boldsymbol{\Gamma}_{h+1}$ is a diagonal matrix of the square root of the diagonal elements of $\boldsymbol{\Xi}_{h+1}$. Similarly, let $\tilde{\boldsymbol{\Lambda}}_{h+1} = \tilde{\boldsymbol{\Gamma}}_{h+1}^{-1} \boldsymbol{\Lambda}_{h+1} \tilde{\boldsymbol{\Gamma}}_{h+1}^{-1}$, where $\boldsymbol{\Gamma}_{h+1}$ is a diagonal matrix of the square root of the diagonal elements of $\boldsymbol{\Lambda}_{h+1}$. Compute $P_{h+1} = \frac{\Phi_3\left(\tilde{\boldsymbol{\Gamma}}_{h+1}^{-1}(\vec{\mathbf{w}}_{h+1} - \boldsymbol{\theta}_{h+1}); \tilde{\boldsymbol{\Lambda}}_{h+1}\right)}{\Phi_2(\boldsymbol{\Gamma}_{h+1}^{-1}(\tilde{\mathbf{w}}_{h+1} - \boldsymbol{\lambda}_{h+1}); \tilde{\boldsymbol{\Xi}}_{h+1})}$.

(7) End for

(8) Return $(P_1 \times \prod_{h=1}^{H-3} P_{h+1})$.

[3] $\boldsymbol{\Xi}_{h+1} = \left(\mathbf{L}_{h+1}^{new}[1:2,1:2]\right)\mathbf{D}_{h+1}^{new}[1:2,1:2]\left(\mathbf{L}_{h+1}^{new}[1:2,1:2]\right)'$ and $\boldsymbol{\Lambda}_{h+1} = \left(\mathbf{L}_{h+1}^{new}[1:3,1:3]\right)\mathbf{D}_{h+1}^{new}[1:3,1:3]\left(\mathbf{L}_{h+1}^{new}[1:3,1:3]\right)'$.

2.3.2. Bivariate Truncation Approaches

2.3.2.1 The Bivariate ME (BME) Method

In this proposed extension of the ME method, we write the MVNCD function as a recursive product of bivariate (conditional) cumulative distributions (BCCCD) (rather than as a recursive product of univariate (conditional) cumulative distributions in the ME method). More specifically, using the notations as earlier, for ($H \geq 2$), even H, and K=Floor(H/2) (that is, H/2 is rounded down to the nearest integer to obtain K):

$$\begin{aligned}\Pr(\boldsymbol{W} < \boldsymbol{w}) &= \Pr(W_1 < w_1, W_2 < w_2) \times \\ &\quad \prod_{k=1}^{K-1} \Pr(W_{2k+1} < w_{2k+1}, W_{2k+2} < w_{2k+2}) \,|\, (W_1 < w_1,\ W_2 < w_2,\ W_3 < w_3, ...,\ W_{2k} < w_{2k}). \\ &\approx \Pr(W_1 < w_1, W_2 < w_2) \times \\ &\quad \prod_{k=1}^{K-1} \Pr(\widetilde{W}_{2k+1} < \widetilde{w}_{2k+1}, \widetilde{W}_{2k+2} < \widetilde{w}_{2k+2}) \,|\, (\widetilde{W}_{2k-1} < \widetilde{w}_{2k-1}, \widetilde{W}_{2k} < \widetilde{w}_{2k}), \\ &\quad \widetilde{W}_r = \frac{\widetilde{Y}_r - E(\widetilde{Y}_r)}{\sqrt{Var(\widetilde{Y}_r)}}, \widetilde{w}_r = \frac{w_r - E(\widetilde{Y}_r)}{\sqrt{Var(\widetilde{Y}_r)}}\end{aligned} \tag{16}$$

If H is odd, there is a singleton added at the end of the latter expression above that takes the form $\Pr(\widetilde{W}_{2K+1} < \widetilde{w}_{2K+1}) \,|\, (\widetilde{W}_{2K-1} < \widetilde{w}_{2K-1}, \widetilde{W}_{2K} < \widetilde{w}_{2K})$. At each step k of this approach, the bivariate distribution of $\widetilde{W}_{2k+1}, \widetilde{W}_{2k+2} \,|\, (\widetilde{W}_{2k-1}, \widetilde{W}_{2k})$ is assumed to be bivariate normal, and the corresponding bivariate (conditional) cumulative distribution function is evaluated using a bivariate normal cumulative distribution function. Formally, the BME algorithm is as follows ($H \geq 2$):

(1) Compute $P_1 = \Phi_2(w_1, w_2 : \rho_{12})$. If H=2, STOP. Return P_{12}.

(2) Compute the two-block LDLT decomposition $\mathbf{LDL}'$ of the original covariance matrix $\boldsymbol{\Sigma}$. For notational convention purposes, set $\boldsymbol{\pi}_1 = E(\widetilde{\mathbf{Y}}_1) = E(\mathbf{X}) = \boldsymbol{\mu}$, $\tilde{\boldsymbol{\Sigma}}_1 = Cov(\tilde{\mathbf{Y}}_1) = Var(\mathbf{X}) = \boldsymbol{\Sigma}$, and $\mathbf{L}_1^{new} = \mathbf{L}, \mathbf{D}_1^{new} = \mathbf{D}$; Compute K=Floor(H/2). If H–2*K=0, set $\widetilde{K} = K - 1$; else set $\widetilde{K} = K$.

(3) For k = 1 to $\widetilde{K}$.

(4) Truncate on the (2k–1) and (2k) elements of $\mathbf{X}$ and obtain $E(\tilde{\mathbf{Y}}_k) = \tilde{\boldsymbol{\mu}}_k$ and $Cov(\tilde{\mathbf{Y}}_k) = \boldsymbol{\Omega}_k$, where $\tilde{\mathbf{Y}}_k = (Y_{2k-1}, Y_{2k})'$. This is done using Equations (4) and (5) in Property 1, with μ_{2k-1} and μ_{2k} (in the computation of $\tilde{\boldsymbol{\mu}}_k$) being replaced by the first two elements of $\boldsymbol{\pi}_k = E(\widetilde{\mathbf{Y}}_k)$, and σ_1^2, σ_2^2, and σ_{12}^2 being replaced by the

elements of the sub-matrix of the first two rows and two columns of $\mathbf{D}_k^{new}$; compute $\boldsymbol{\pi}_{k+1} = E(\widetilde{\mathbf{Y}}_{k+1}) = \boldsymbol{\pi}_k[3:H-2k+2] + \mathbf{L}_k^{new}[3:H-2k+2,1:2]\,(\tilde{\boldsymbol{\mu}}_k - \boldsymbol{\pi}_k[1:2])$ using Property 3.

(5) Implement a two-block LDLT decomposition of $\tilde{\boldsymbol{\Sigma}}_{k+1} = Cov(\tilde{\mathbf{Y}}_{k+1})$ using a call to the LDLT rank-2 updating algorithm with inputs $\{\mathbf{L}_k, \mathbf{D}_k, \boldsymbol{\Omega}_{\mathbf{k}}\}$ to obtain $\mathbf{L}_{k+1}^{new}$ and $\mathbf{D}_{k+1}^{new}$.

(6) If $H < 2k+2$, obtain $\boldsymbol{\theta}_{k+1} = E(Y_{2k+1})$ as the (only) entry of the vector $\boldsymbol{\pi}_{k+1} = E(\widetilde{\mathbf{Y}}_{k+1})$, and $Var(Y_{2K+1})$ as the only element of $\mathbf{D}_{\mathbf{k+1}}^{new}$, and compute $P_{k+1} = \Phi\left(\frac{w_{2K+1} - E(Y_{2K+1})}{\sqrt{Var(Y_{2K+1})}}\right)$. If $H \geq 2k+2$, obtain $\boldsymbol{\theta}_{k+1} = (E(Y_{2k+1}), E(Y_{2k+2}))'$ as the first two entries of the vector $\boldsymbol{\pi}_{k+1} = E(\widetilde{\mathbf{Y}}_{k+1})$. Let $\widetilde{\mathbf{w}}_{k+1} = (w_{2k+1}, w_{2k+2})'$, and let $\boldsymbol{\Xi}_{k+1}$ be the sub-matrix of the first two rows and two columns of $\mathbf{D}_{k+1}^{new}$.[4] Let $\widetilde{\boldsymbol{\Xi}}_{k+1} = \boldsymbol{\Gamma}_{k+1}^{-1}\boldsymbol{\Xi}_{k+1}\boldsymbol{\Gamma}_{k+1}^{-1}$, where $\boldsymbol{\Gamma}_{k+1}$ is a diagonal matrix of the square root of the diagonal elements of $\boldsymbol{\Xi}_{k+1}$. Compute $P_{k+1} = \Phi_2(\boldsymbol{\Gamma}_{k+1}^{-1}(\widetilde{\mathbf{w}}_{k+1} - \boldsymbol{\theta}_{k+1}); \widetilde{\boldsymbol{\Xi}}_{k+1})$.

(7) End for

(8) Return ($P_1 \times \prod_{k=1}^{\tilde{K}} P_{k+1}$).

Trinh and Genz (2015) also develop a bivariate conditioning approximation in their algorithm 3.2. However, again, as in the case of the (univariate) ME method, they ignore the $Cov(\widetilde{\mathbf{Y}}_{k+1})$ term at each k-step, and simply retain the original variance $\boldsymbol{\Sigma}$ without any updating at each *k*-step. In other words, there is no computation of $Cov(\widetilde{\mathbf{Y}}_{k+1})$ in step 4 of our proposed approach, and step 5 of our algorithm above is completely ignored. The result is again an overestimation in the MVNCD approximation relative to our bivariate ME method, and larger errors relative to our proposed bivariate ME method.

[4] Note that, because of the nature of the LDLT-block decomposition, it will be true that the sub-matrix of the first two rows and first two columns of $\mathbf{D}_{k+1}^{new}$ will be exactly the same as the sub-matrix of the first two rows and first two columns of $\tilde{\boldsymbol{\Sigma}}_{k+1} = Cov(\tilde{\mathbf{Y}}_{k+1})$.

2.3.2.2. The Two-Variate Bivariate Screening (TVBS) Method

At each step k of the proposed bivariate ME method, the bivariate distribution of $\widetilde{W}_{2k+1}, \widetilde{W}_{2k+2} \mid (\widetilde{W}_{2k-1}, \widetilde{W}_{2k})$ is assumed to be bivariate normal. However, a much more appropriate assumption is to acknowledge that this is a kind of multivariate skew normal distribution, or, in Kim and Kim's (2015) terminology, a two-variate bivariate-screened distribution. Accordingly, the corresponding conditional cumulative distribution function at each step is better represented as a ratio of a four-variate normal cumulative distribution function and a two-variate normal cumulative distribution function (CDF). But since a four-variate normal CDF can be time-consuming, we approximate the four-variate normal CDF very fast and accurately by taking the trivariate cumulative CDF of the first three variates (P_{123}), truncating on the first two variates using Property 1 to obtain the expected values and variance-covariance matrix of the third and four variates conditioned on the truncation on the first two variates, and then approximating the cumulative distribution function of the last variate given the third variate as a skew normal using step 6 of the one-variate univariate screening approach. The formal algorithm is as follows (H ≥ 2), where we will use the notation $\widetilde{\Phi}_4(.)$ to represent the accurate approximation just described for $\Phi_4(.)$.

(1) If *H*=2, return $P_1 = \Phi_2(w_1, w_2 : \boldsymbol{\Sigma})$. STOP. If *H*=3, return $P_1 = \Phi_3(w_1, w_2, w_3 : \boldsymbol{\Sigma})$. STOP. If *H*=4, return $P_1 = \widetilde{\Phi}_4(w_1, w_2, w_3, w_4 : \boldsymbol{\Sigma})$. STOP. If *H*>4, compute $P_1 = \tilde{\Phi}_4(w_1, w_2, w_3, w_4 : \tilde{\boldsymbol{\Lambda}}_1)$. $\tilde{\boldsymbol{\Lambda}}_1$ is the marginal quadrivariate correlation sub-matrix of the first four rows and first four columns of $\boldsymbol{\Sigma}$.

(2) Compute the two-block LDLT decomposition $\mathbf{LDL}'$ of the original covariance matrix $\boldsymbol{\Sigma}$. For notational convention purposes, set $\boldsymbol{\pi}_1 = E(\widetilde{\mathbf{Y}}_1) = E(\mathbf{X}) = \boldsymbol{\mu}$, $\tilde{\boldsymbol{\Sigma}}_1 = Cov(\tilde{\mathbf{Y}}_1) = Var(\mathbf{X}) = \boldsymbol{\Sigma}$, and $\mathbf{L}_1^{new} = \mathbf{L}, \mathbf{D}_1^{new} = \mathbf{D}$; Compute *K*=Floor(*H*/2). If *H*–2**K*=0, set $\widetilde{K} = K - 1$; else set $\widetilde{K} = K$.

(3) For $k = 1$ to $\widetilde{K} - 1$;

(4) Same as in our proposed bivariate ME method.

(5) Implement a two-block LDLT decomposition of $\tilde{\boldsymbol{\Sigma}}_{k+1} = Cov(\tilde{\mathbf{Y}}_{k+1})$ using a call to the LDLT rank-2 updating algorithm with inputs $\{\mathbf{L}_k, \mathbf{D}_k, \boldsymbol{\Omega}_k\}$ to obtain $\mathbf{L}_{k+1}^{new}$ and $\mathbf{D}_{k+1}^{new}$.

(6) Obtain $\boldsymbol{\lambda}_{k+1} = (E(Y_{2k+1}), E(Y_{2k+2}))'$ as the first two entries of the vector $\boldsymbol{\pi}_{k+1} = E(\widetilde{\mathbf{Y}}_{k+1})$. Let $\boldsymbol{\Xi}_{k+1}$ be the sub-matrix of the first two rows and first two columns of $\mathbf{D}_{k+1}^{new}$ (equivalently, this is the covariance sub-matrix of the first two rows and two columns of $\tilde{\boldsymbol{\Sigma}}_{k+1}$). Let $\tilde{\boldsymbol{\Xi}}_{k+1} = \boldsymbol{\Gamma}_{k+1}^{-1}\boldsymbol{\Xi}_{k+1}\boldsymbol{\Gamma}_{k+1}^{-1}$, where $\boldsymbol{\Gamma}_{k+1}$ is a diagonal matrix of the square root of the diagonal elements of $\boldsymbol{\Xi}_{k+1}$. If H<2*k+4, obtain $\boldsymbol{\theta}_{k+1} = (E(Y_{2k+1}), E(Y_{2k+2}), E(Y_{2k+3}))'$ as the three entries of the vector $\boldsymbol{\pi}_{k+1} = E(\widetilde{\mathbf{Y}}_{k+1})$. Let $\boldsymbol{\Lambda}_{k+1}$ be the covariance matrix $\tilde{\boldsymbol{\Sigma}}_{k+1}$, which can be obtained as $\mathbf{L}_{k+1}^{new}\mathbf{D}_{k+1}^{new}(\mathbf{L}_{k+1}^{new})'$. Let $\widetilde{\boldsymbol{\Lambda}}_{k+1} = \widetilde{\boldsymbol{\Gamma}}_{k+1}^{-1}\boldsymbol{\Lambda}_{k+1}\widetilde{\boldsymbol{\Gamma}}_{k+1}^{-1}$, where $\widetilde{\boldsymbol{\Gamma}}_{k+1}$ is a diagonal matrix of the square root of the diagonal elements of $\boldsymbol{\Lambda}_{k+1}$. Let $\widetilde{\mathbf{w}}_{k+1} = (w_{2k+1}, w_{2k+2})'$ and $\vec{\mathbf{w}}_{k+1} = (w_{2k+1}, w_{2k+2}, w_{2k+3})'$. Compute $P_{k+1} = \dfrac{\Phi_3\left(\widetilde{\boldsymbol{\Gamma}}_{k+1}^{-1}(\vec{\mathbf{w}}_{k+1} - \boldsymbol{\theta}_{k+1}); \widetilde{\boldsymbol{\Lambda}}_{k+1}\right)}{\Phi_2(\boldsymbol{\Gamma}_{k+1}^{-1}(\widetilde{\mathbf{w}}_{k+1} - \boldsymbol{\lambda}_{k+1}); \tilde{\boldsymbol{\Xi}}_{k+1})}$. If H≥2*k+4, obtain $\boldsymbol{\theta}_{k+1} = (E(Y_{2k+1}), E(Y_{2k+2}), E(Y_{2k+3}), E(Y_{2k+4}))'$ as the first four entries of the vector $\boldsymbol{\pi}_{k+1} = E(\widetilde{\mathbf{Y}}_{k+1})$. Let $\boldsymbol{\Lambda}_{k+1}$ be the covariance sub-matrix of the first four rows and four columns of $\tilde{\boldsymbol{\Sigma}}_{k+1}$, computed as $\mathbf{L}_{k+1}^{new}[1:4,1:4]\mathbf{D}_{k+1}^{new}[1:4,1:4](\mathbf{L}_{k+1}^{new}[1:4,1:4])'$. Let $\widetilde{\boldsymbol{\Lambda}}_{k+1} = \widetilde{\boldsymbol{\Gamma}}_{k+1}^{-1}\boldsymbol{\Lambda}_{k+1}\widetilde{\boldsymbol{\Gamma}}_{k+1}^{-1}$, where $\widetilde{\boldsymbol{\Gamma}}_{k+1}$ is a diagonal matrix of the square root of the diagonal elements of $\boldsymbol{\Lambda}_{k+1}$. $\boldsymbol{\pi}_{h+1} = E(\widetilde{\mathbf{Y}}_{h+1})$. Let $\widetilde{\mathbf{w}}_{k+1} = (w_{2k+1}, w_{2k+2})'$ and $\vec{\mathbf{w}}_{k+1} = (w_{2k+1}, w_{2k+2}, w_{2k+3}, w_{2k+4})'$. Compute $P_{k+1} = \dfrac{\widetilde{\Phi}_4\left(\widetilde{\boldsymbol{\Gamma}}_{k+1}^{-1}(\vec{\mathbf{w}}_{k+1} - \boldsymbol{\theta}_{k+1}); \widetilde{\boldsymbol{\Lambda}}_{k+1}\right)}{\Phi_2(\boldsymbol{\Gamma}_{k+1}^{-1}(\widetilde{\mathbf{w}}_{k+1} - \boldsymbol{\lambda}_{k+1}); \tilde{\boldsymbol{\Xi}}_{k+1})}$.

(7) End for

(8) Return ($P_1 \times \prod_{k=1}^{K-1} P_{k+1}$).

3. THE EVALUATION DESIGN

We tested the multiple methods in two ways in this paper. The first was to evaluate the accuracy of computing the MVNCD function directly for different dimensions (H = 5, 7, 10, 12, 15, 18, and 20), and the second was to examine the recovery ability of underlying MNP parameters with different numbers of alternatives (H + 1 = 6, 11, 16, 21) that lead to MVNCD evaluations of H dimensions (H = 5, 10, 15, and 20; an MNP with H+1 alternatives requires the evaluation of an H-dimensional MVNCD function).

3.1. The Direct MVNCD Function Evaluation

For the first set of experiments, for each H value, we first generate 1000 random positive-definite covariance matrices, based on a simple factoring approach, as $\mathbf{C} = \mathbf{R}\mathbf{R}' + \delta \times diag(\mathbf{ru})$, where $\mathbf{R}$ is

a matrix of $H \times H$ random univariate standard normal variates, and $diag(\mathbf{ru})$ represents a diagonal matrix with the $H \times 1$ vector **ru** of standard uniform random variates on the diagonal. δ is a scalar that determines the relative magnitude of the diagonal elements relative to the non-diagonal elements. Finally, the positive definite random correlation matrix is obtained from the generated positive definite covariance matrix **C**. In this set-up, the higher the value of δ, the lower are the correlations in the corresponding correlation matrix. We use two different values of δ; one with a value of 10, generating 500 correlation matrices with low correlations (with the correlations ranging from -0.8566 to 0.8085), and another with a value of zero, generating another 500 correlation matrices with high correlations (with the correlations ranging from -0.9634 to 0.9781).[5] For half of the 500 matrices from each of the low correlation case and high correlation cases, the upper integration limits were uniformly drawn (separately and independently for each of the H dimensions) from $[0,+\sqrt{H}]$.[6] This produced "exact" MVNCD values (see next paragraph for how the "exact" MVNCD value was computed) that ranged from 0.0044 to 0.8868 (overall mean of 0.2502) in our experiments across all the H values (with the range being 0.0622 to 0.8868 for H=5 with a mean of 0.3933, and 0.0044 to 0.8479 for H=20 with a mean of 0.1519). Next, to also have representation at the lower values of the MVNCD function, we generated another set of 500 upper integration limits (corresponding to the other half of the 1000 correlation matrices) now drawn from $[-\sqrt{H}/2,+\sqrt{H}]$ as $\frac{3}{2}\sqrt{H} * rndu[0,1] - \frac{\sqrt{H}}{2}$. This produced "exact" MVNCD values that ranged from 0^+ to 0.8988 (overall mean of 0.0273) across all the H values (with the range being 0^+ to 0.8988 for H=5 with a mean of 0.1143, and shrinking rapidly to 0^+ to 0.0356 for H=20 with a mean of 0.0003). Taken together, we consider four sets of 250 MVNCD evaluations each, corresponding to (a) low correlations, high MVNCD values, (b) low correlations, low MVNCD values, (c) high correlations, high MVNCD values, and (d) high correlations, low MVNCD values.

[5] The random generation of correlations across all H values shows extreme values that do not differ much between the low and high correlation cases, but the overall correlation magnitudes depend on the value of H and do vary quite a bit between the low and high correlation cases. For example, for the case of H=5, 74.2% of the absolute values of the generated correlations lie between 0 and 0.25 for the low correlation case, relative to only 35.8% for the high correlation case. On the other hand, less than 0.1% of the absolute values of the generated correlations lie between 0.75 and 1 for the low correlation case, relative to almost 9% for the high correlation case.

[6] The reason we draw uniform variates from $[0,+\sqrt{H}]$ for the upper integration limits (so that the upper integration limits generally are higher for high H) is because, as H increases, the MVNCD value gets closer and closer to zero for a given upper integration limit. By increasing the upper integration limit as H increases, we are allowing for a better distribution of the MVNCD value in the $\{0,1\}$ range.

The 1000 evaluations cover the range of MVNCD values from 0 to 1, as well as low and high correlation patterns.

The "exact" evaluation of each of the 1000 MVNCD functions was undertaken using the CDFMVNe function of the GAUSS matrix programming language, up to an accuracy of 1e-6 (i.e., 0.000001). The various methods were compared against this exact value by computing the mean absolute error or MAE (mean across the 1000 evaluations of the absolute difference between the MVNCD from the method relative to the "exact" value), the mean absolute percentage error or MAPE, the percentage of the 1000 evaluations in which the error was over 0.005, and the percentage of evaluations in which the percentage error exceeded two percent. The methods tested, for each of the H values of 5, 7, 10, 12, 15, 18, and 20, include the following: the GHK-Halton 500 (GHK-Halton with 500 draws), the GHK-Halton 10000 (GHK-Halton with 10,000 draws), the SSJ1, the SSJ10, the TGME method, the true ME method, the OVUS method, the OVBS method, the TGBME method, the BME method, and the TVBS method (for a total of 11 approaches). In addition to the metrics above, we also computed the total time (in seconds) for evaluation of the 1000 MVNCD functions for each method. For the ME method, as well as the proposed new methods and the TG methods, we use an optimally ordered version based on the decreasing range of the abscissae (we also considered a more refined version of ordering based on the outermost integral variables having the smallest expected values, as suggested by Gibson et al., 1994 and implemented in Trinh and Genz, 2015; but this refined version provided little to no gains in accuracy in our experiments relative to the more simple reordering used here).

3.2. The MNP Model Set-up

As discussed earlier, the estimation of an MNP with H+1 alternatives requires the evaluation of an H-dimensional MVNCD function. While traditional simulation procedures have been primarily used for evaluation of these MVNCD functions, Bhat (2011) proposed the analytic approximation-based maximum approximate composite marginal likelihood (MACML) approach. In doing so, Bhat underscores the convergence and computational cost problems underlying the traditional simulation-based approaches to maximum likelihood inference estimation.[7] Bhat and colleagues

[7] In particular, the computational cost to ensure desirable asymptotic properties of the simulation-based estimators (consistency, efficiency, and asymptotic normality) can be prohibitive and literally infeasible (in the context of the computation resources available and the time available for estimation) as the number of dimensions of integration increases, because the accuracy of simulation techniques is known to degrade rapidly at medium-to-high dimensions,

(see Bhat, 2015 for a review) have also applied this MACML approach for MNP models with spatial/social dependence models, making use of the CML inference approach to dramatically reduce the dimensionality of expressions during estimation from the order of the number of individuals times the number of alternatives (in the traditional maximum likelihood approach) to the number of alternatives (in the CML approach).

In this paper, we consider a simple non-spatial MNP setting and evaluate alternative MVNCD function evaluation approaches in estimating the MNP model. A challenge in doing so is to compare parameter estimates using the alternative approaches against the true estimates, given a sample. In this paper, we use a key insight that allows us to obtain the true estimates of the underlying MNP parameters very accurately and quickly for a particular setting of the MNP, given a data sample and regardless of the dimensionality involved in the resulting MVNCD evaluation. The insight is the reverse of that used by Bhat (2003) when his objective there was to test simulation techniques in a mixed discrete choice setting.

To make things clear, consider the following set-up for an MNP model in which the utility that an individual q associates with alternative h ($h = 1, 2, ..., H+1$) is written as:

$$U_{qh} = \boldsymbol{b}'\boldsymbol{x}_{qh} + \gamma_q z_{qh} + \xi_{qh};\ \gamma_q = c + \widetilde{\gamma}_q,\ \widetilde{\gamma}_q \sim N(0, \sigma^2), \tag{17}$$

where $\boldsymbol{x}_{qh}$ is an $(E \times 1)$-column vector of exogenous attributes, $\boldsymbol{b}$ is a fixed $(E \times 1)$-column vector of corresponding coefficients, z_{qh} is a single exogenous variable corresponding to individual q and alternative h, and γ_q is a random coefficient (capturing unobserved heterogeneity or response sensitivity variation across individuals to the variable z_{qh}) that is a realization from a normal distribution with mean c and variance τ^2 [$\text{cov}(\gamma_q, \gamma_{q'}) = 0\ \forall q, q'$]. ξ_{qh} is assumed to be an independently and identically distributed (across alternatives and across individuals) normal error term with a variance of 0.5. This variance has to be preset for identification purposes. Let $\boldsymbol{\xi_q} = (\xi_{q1}, \xi_{q2}, ..., \xi_{q,H+1})'$ ($(H+1) \times 1$ vector). Then, $\boldsymbol{\xi_q} \sim MVN_{H+1}(\boldsymbol{0}, 0.5 * \boldsymbol{IDEN}_{H+1})$, where

and the simulation noise increases substantially (leading to convergence problems too during estimation). More generally, Patil et al. (2017) have recently shown that the routinely invoked good asymptotic properties of simulation-based estimators do not translate to a lack of bias and low finite sample efficiency in the range of finite sample sizes and numbers of simulation draws typically used in consumer choice estimations. On the other hand, while analytic approximation-based estimators of the MACML type do not necessarily bring with them the theoretically appealing asymptotic properties (see Batram and Bauer, 2016), they can be far superior to simulation-based estimators in routine consumer choice analyses in the typical sample sizes available for estimation.

$\boldsymbol{IDEN}_{H+1}$ is an identity matrix of size (H+1). ξ_{qh} is also assumed to be independent of γ_q. Also, collect all the parameters to be estimated into a vector $\boldsymbol{\theta} = (\boldsymbol{b}', c, \tau^2)'$.

In the current paper, for each number of alternatives ($H + 1 = 6$, 11, 16, and 21), we generate a sample of 3000 observations corresponding to 3000 individuals. We consider two variables in the $\boldsymbol{x}_{qh}$ vector: a continuous variable and a dummy variable taking the values 0/1. We also consider a continuous variable for z_{qh}. Overall, then, we have a total of two continuous variables and a single dummy variable in the experimental design for MVNCD evaluation ability in the MNP estimation.[8] The values for the continuous variable (x_{qh1}) in the $\boldsymbol{x}_{qh}$ vector are drawn for the first half of alternatives (the first three in the six-alternative case, the first six in the 11-alternative case, the first eight in the 16-alternative case, and the first 11 in the 21-alternative case) from a standard univariate normal distribution, while the corresponding values for the remaining alternatives are drawn from a univariate normal distribution with mean 0.5 and standard deviation of 1. The parameter b_1 on x_{qh1} is specified to be one across all alternatives. The procedure is exactly reversed for z_{qh}, with the values for the first half of alternatives drawn from a univariate normal distribution with mean 0.5 and standard deviation of one and the values for the remaining alternatives drawn from a standard univariate normal distribution. The parameter on z_{qh} is drawn from a normal distribution with a mean (c) of -0.5 and variance (τ^2) of 1 (that is, 3000 realizations are drawn from the normal distribution, and applied to z_{qh} for each alternative across the many tests with different numbers of all alternatives). For the dummy variable (x_{qh2}) in $\boldsymbol{x}_{qh}$, we treat this as an individual-specific variable (that does not vary across alternatives). To construct this dummy variable, 3000 independent values are drawn from the standard uniform distribution. If the value drawn is less than 0.5, the value of '0' is assigned for the dummy variable. Otherwise, the value of '1' is assigned. The dummy variable values, once generated, are held the same across the many tests with different numbers of alternatives. The coefficients on this dummy variable are specified to be 0 for the first half of alternatives (as defined earlier) and 0.75 for the second half. Thus, a

[8] We do not include alternative-specific constants in our design because this would add many more parameters to the model. For example, in the 21 alternative case, there would be 20 additional alternative-specific constants. Our focus here is on the ability to evaluate the MVNCD function accurately as part of the MNP estimation, and so we choose to keep the number of parameters embedded within the MVNCD function in the MNP model to a small number.

single parameter b_2 is to be estimated for the dummy variable (note that, in estimation, we constrain the coefficients to be zero for the first half of alternatives, and estimate a single parameter that is constrained to be equal across the second half of alternatievs). The parameters to be estimated from the data generating process above correspond to $\theta = (b_1 = 1, b_2 = 0.75,\ c = -0.5, \tau^2 = 1)'$. However, because we are using a single sample to estimate the parameters, the actual estimates of these parameters (even if exactly determined) on the sample generated will not be identical to the above. In the current paper, we are able to determine the actual true estimates on the specific sample generated for each combination of correlation intensity (low and high) and number of alternatives, as discussed next. The ability of the approximation methods to recover these parameters specific to the samples generated is tested against the true estimates for each sample.

With the set-up as above, the likelihood function for the estimation of the parameters can be computed in one of two ways: (a) the easy "exact" way using a mixed probit framework that exploits the independence across utilities of alternatives conditional on γ_q or (b) the "approximate" way using a traditional multinomial probit (MNP) framework. These two methods are discussed in the two subsequent sections.

3.2.1. The Mixed Probit Framework

Let the individual *q* choose the m_qth alternative. Define $\tilde{\boldsymbol{x}}_{qhm_q} = \boldsymbol{x}_{qh} - \boldsymbol{x}_{qm_q}$, $\tilde{z}_{qhm_q} = z_{qh} - z_{qm_q}$, and $\tilde{V}_{qhm_q} = \boldsymbol{b}'\boldsymbol{x}_{qhm_q} + c\,\tilde{z}_{qhm_q}$. Because the individual chose the m_qth alternative, $(U_{qh} < U_{qm_q})\ \forall\ h \neq m_q$. That is, $\varepsilon_{qh} < -\tilde{V}_{qhm_q} - \tilde{\gamma}_q \tilde{z}_{qhm_q} + \varepsilon_{qm_q}\ \forall\ h \neq m_q$. Using the independence assumption across the error terms ε_{qh}, we can write individual *q*'s likelihood function for the choice of the m_qth alternative, conditional on a given realization of $\tilde{\gamma}_q$ and ε_{qm_q}, as

$$L_q(\theta) \mid \tilde{\gamma}_q, \varepsilon_{qm_q} = \prod_{h \neq m_q} \left[\Phi\left[\sqrt{2}\left(-\tilde{V}_{qhm_q} - \tilde{\gamma}_q \tilde{z}_{qhm_q} + \varepsilon_{qm_q}\right)\right]\right]$$

Then, the unconditional likelihood function for individual *q* is given by:

$$L_q(\theta) = \int_{\tilde{\vartheta}=-\infty}^{\infty} \left\{ \int_{\tilde{\lambda}=-\infty}^{\infty} \left(L_q(\theta) \mid \tilde{\gamma}_q, \varepsilon_{qm_q}\right) f_{\varepsilon_{qm_q}}(\tilde{\lambda}) d\tilde{\lambda} \; f_{\tilde{\gamma}_q}(\tilde{\vartheta}) \right\}$$

$$= \int_{\vartheta=-\infty}^{\infty} \left\{ \int_{\lambda=-\infty}^{\infty} \prod_{h \neq m_q} \Phi\left[-\sqrt{2}\left(\tilde{V}_{qhm_q} + (\tau\,\vartheta)\tilde{z}_{qhm_q}\right) + \lambda \right] \phi(\lambda) d\lambda\, \phi(\vartheta) d\vartheta \right\}$$

The likelihood function above involves only a two-dimensional integral regardless of the number of alternatives H+1. The two-dimensional integral can be evaluated very accurately using traditional simulation techniques. Here, we evaluated the integral using 35,000 pseudo-random draws (there was little change in the parameter estimates with as few as 10,000 pseudo-random draws, but we increased to 35,000 draws to get "exact" true estimates for the $\boldsymbol{\theta}$ parameter vector and the standard error of its elements).

3.2.2. The Traditional MNP Estimation

For the traditional MNP estimation, let $\mathbf{U}_q = (U_{q1}, U_{q2}, ..., U_{qH+1})'$ $((H+1)\times 1$ vector), $\boldsymbol{x_q} = (\boldsymbol{x_{q1}}, \boldsymbol{x_{q2}}, \boldsymbol{x_{q3}}, ..., \boldsymbol{x_{q,H+1}})'$ $((H+1)\times E$ matrix), $\boldsymbol{z_q} = (z_{q1}, z_{q2}, z_{q3}, ..., z_{q,H+1})'$ $((H+1)\times 1$ vector), $\mathbf{V}_q = \boldsymbol{x}_q \boldsymbol{b} + \boldsymbol{z}_q \gamma$ $((H+1)\times 1$ vector), $\boldsymbol{\Omega}_q = \boldsymbol{z}_q(\sigma^2)\boldsymbol{z}_q' + \mu^2 \boldsymbol{IDEN}_{H+1}$ $((H+1)\times(H+1)$ matrix$)$. Then, we may write, in matrix notation, $\mathbf{U}_q = \mathbf{V}_q + \boldsymbol{z}_q\tilde{\gamma} + \boldsymbol{\xi}_q$ and $\mathbf{U}_q \sim MVN_{H+1}(\mathbf{V}_q, \boldsymbol{\Omega}_q)$. Also, let $\mathbf{u}_q = (u_{q1}, u_{q2}, \ldots, u_{q,H+1})'$ $(h \neq m_q)$ be an H×1 vector, where, as earlier, m_q is the actual observed choice of individual q, and $u_{qh} = U_{qh} - U_{qm_q}$ $(h \neq m_q)$. Then, $\mathbf{u}_q < \mathbf{0}_H$, because alternative m_q is the chosen alternative by individual q.

To develop the likelihood function, define $\mathbf{M}_q$ as an identity matrix of size H with an extra column of '-1' values added at the m_q^{th} column (thus, $\mathbf{M}_q$ is a matrix of dimension $(H \times (H+1))$. Then, $\mathbf{u}_q$ is distributed as follows: $\mathbf{u}_q \sim MVN_H(\boldsymbol{B_q}, \boldsymbol{\Xi_q})$, where $\boldsymbol{B_q} = \mathbf{M}_q \mathbf{V}_q$ and $\boldsymbol{\Xi_q} = \mathbf{M}_q \boldsymbol{\Omega_q} \mathbf{M}_q'$. Let $\boldsymbol{\omega}_{\boldsymbol{\Xi}_q}$ be the diagonal matrix of standard deviations of $\boldsymbol{\Xi}_q$. Using the usual notations as described earlier, the likelihood contribution of individual q is as below:

$$L_q(\boldsymbol{\theta}) = \Phi_H(\boldsymbol{\omega}_{\boldsymbol{\Xi}_q}^{-1}(-\boldsymbol{B}_q), \boldsymbol{\Xi}_q^{\,*}), \tag{18}$$

where $\boldsymbol{\Xi}_q^{\,*} = \boldsymbol{\omega}_{\boldsymbol{\Xi}_q}^{-1} \boldsymbol{\Xi}_q \, \boldsymbol{\omega}_{\boldsymbol{\Xi}_q}^{-1}$.

The likelihood function above requires the evaluation of an H-dimensional MVNCD function, and forms the basis for comparing the performance of alternative computation methods in estimating the underlying parameters when compared to the true estimates of the parameter vector $\boldsymbol{\theta}$ obtained from the mixed probit approach in the way we have set things up. For each number of alternatives, 11 estimations are undertaken corresponding to each of the 11 analytic MVNCD approximation methods. All the methods were implemented using the GAUSS matrix programming language. The approximated log-likelihood function and the gradients of the approximated log-likelihood function with respect to relevant parameters were coded.

The true estimates of the parameters from the mixed probit estimation served as the benchmark to compare the performances of the alternate MVNCD approximation methods. The performance evaluation of each method was based on five criteria: (a) ability to recover the true estimates of the model parameters, (b) ability to recover the true sampling standard error of the estimator, (c) ability to estimate the overall log-likelihood function accurately, (d) ability to reproduce individual (*i.e.*, observation-specific) likelihood function values, and (e) ability to replicate the logarithm of the individual likelihood function. For each of these criteria, the evaluation of the proximity of the estimated parameter values from the different MVNCD approximation methods and the true estimates was based on the mean absolute percentage error (MAPE). The time to convergence using each method is also obtained.

4. THE RESULTS

4.1. The Individual MVNCD Evaluation Results

Table 1 provides the overall results across the 1000 MVNCD evaluations. Across the many methods, the GHK-500 (the first numeric column in Table 1) starts off well with H=5, with an MAE of 0.00061 (see first numeric row under GHK-500) and an MAPE value of 0.92, but the performance deteriorates rapidly as the number of dimensions increase. At H=20, the MAE for GHK-500 increases to 0.00124, with a corresponding MAPE value of 18.53%. Also, the percentage of MVNCD evaluations with an MAPE value greater than 2% for the GHK-500 increases from 4.3% for H=5 to 63.3% for H=20. Concomitant with this degradation in accuracy is a rapid increase in computational time, from 0.071 seconds on average for each MVNCD computation at H=5 to a seven-fold increase to 0.458 seconds at H=20. The GHK-10000 (the second numeric column in Table 1), as expected, has a much better performance than the GHK-

500, with the MAE-based accuracy increasing by one digit or about a ten-fold improvement across all H values. However, again, there is rapid degradation as one goes from H=5 to H=20, with the MAPE increasing from 0.14% at H=5 to 10.76% at H=20 and the percentage of MVNCD evaluations with an MAPE value greater than 2% increasing from 0.9% at H=5 to 34.1% at H=20. As importantly, the computational time for the GHK-10000 explodes with an increase in H, from close to 1 second for H=5 to 11.39 seconds for H=20.

The SSJ1 does not do as well as the GHK-500 at low H values (H = 5, 7, and 10), but, remarkably, does better than the GHK-500 at high H values (H = 12, 15, 18, and 20) in terms of MAE and about the same as the GHK-500 at these high H values in terms of other metrics. However, the computation time of SSJ1, which is about half the time needed for GHK-500 at low H values, becomes almost comparable to that of GHK-500 at the high H values. The SSJ10 method, relative to the SSJ1 method, improves accuracy at the low H values, but this improvement decreases as we move into the high H ranges. At the same time, as with the case with GHK-10000, the computation time is of the order of seconds (3.093 seconds, on average) for the SSJ10 at H=20. Overall, among the four SSJ and GHK methods, and based on a combination of accuracy and computational time, it appears that the GHK-500 would be the preferred method until H=10, but the SSJ1 method becomes the preferred method at higher H values.

Moving to the four univariate truncation methods, the TGME method does very poorly (and by an order of magnitude) relative to the SSJ and GHK methods for all H values. The ME method, as expected, performs substantially better than the TGME method for the reasons already discussed conceptually earlier. However, the ME method does not do as well as the GHK-500 method for H=5 and 7. But, beyond H=7, while the GHK-500 degrades rapidly, the ME continues to do well, and surpasses the GHK-500 on all metrics, even though it takes a little more time to compute than the GHK-500. At these higher H values, the ME is also superior to the SSJ1, though it does take about twice the time to compute as the SSJ1. The implication so far then is that, between the SSJ, GHK, and ME methods, the GHK-500 appears to be the preferred method for MVNCD evaluation until H=7, after which the ME comes out on top.

Among the univariate truncation methods, the trend is clear. The OVUS is superior to the ME, and the OVBS is superior to the OVUS, for every H value. Also, there is no substantial difference in computation times across the ME, OVUS, and OVBS approaches. As importantly, the OVBS is superior to the GHK-500 even for H=5 in terms of MAE, and very comparable to

GHK-500 on other metrics for H=5. At H=7 and beyond, it is clearly the better approach relative to GHK-500. Interestingly, the ME, OVUS, and OVBS methods all provide better accuracy than even the GHK-10000 draws, and at about 12-14 times faster speed, at H=20. Based on accuracy and computation time, among all the GHK, SSJ, and univariate truncation schemes, the OVBS comes out clearly on top across all H values.

The last column panel of Table 1 provides the results for the three bivariate truncation methods. Again, the TGBME does very poorly here, even if marginally better than the TGME. On all the metrics, the TGME and the TGBME methods provide the worst results, a reflection of completely ignoring the variance component of truncation elements. The BME does better than the ME for all H values, and at lesser computation cost (while the bivariate truncations involve more computation time in computing the moments of a bivariate truncated vector of elements, as discussed in Section 2, it also has fewer iterations or "flops" because it races faster to cover the dimensionality H of variables; see step 3 in the univariate and truncation algorithms). Our special LDLT-based implementation of the algorithms also helps in ensuring that there is no substantial speed reduction within each iteration because of using a bivariate truncation scheme relative to a univariate truncation scheme. Between the BME and the OVBS methods (which came out to be the best in all the non-bivariate truncation approaches), the OVBS seems to do marginally better until H=12, after which the two are similar, with the BME faring slightly better on most accuracy metrics. The BME is, however, about twice as fast as the OVBS. Finally, between the TVBS and the BME methods, the TVBS is clearly better on all metrics for all H values (except for H=20, where there is literally no difference between the two). Further, the performance of the TVBS is as good or better than the OVBS across the board, and the TVBS faster than the OVBS. Overall, the TVBS is the best in terms of accuracy and speed in evaluating individual MVNCD functions.

To further investigate performance of the different methods, we also examined their performances in each of four sets of cases corresponding to (a) low correlations, high MVNCD values, (b) low correlations, low MVNCD values, (c) high correlations, high MVNCD values, and (d) high correlations, low MVNCD values. Some important insights from this exercise are as follows (the results in the same form as Table 1, but separately for each of the four cases, are available at http://www.caee.utexas.edu/prof/bhat/ABSTRACTS/LDLT/OnlineSupplement.pdf). First, all methods do very well in the case of low correlations and high MVNCD values. This is not surprising, because all methods will provide the exact values when the elements are fully

independent. Second, the algorithms generally do well even for the second set of low correlations and low MVNCD values, though the %MAPE metric starts getting high because of the extremely low exact values of the MVNCD function. But, in addition to the TG and the TGBME methods that always do poorly, the SSJ and GHK-500 also perform very poorly relative to the ME, OVUS, OVBS, BME, and TVBS methods in this second set (much more so than in Table 1; for example, with H=20, the percentage of MVNCD evaluations with an MAPE value higher than 2% for the SSJ and GHK-500 approaches are of the order of 85-90% compared to about 30% for the other approaches). Third, the results for the last two sets with high correlations also show good performances in terms of MAE, though the MAE values are higher for the third set relative to the first and for the fourth set relative to the second. Again, relative to the ME, OVUS, OVBS, BME, and TVBS approaches, the SSJ and GHK-500 approaches perform much more poorly than in the general results of Table 1 for the fourth set of high correlation and low MVNCD values. Finally, overall, the same relative performances of the algorithms are observed as discussed earlier across all sets for each individual set of 250 MVNCD evaluations. Of course, in general, it is also difficult to know a priori whether the correlations are going to be high or low, or whether the truncation of the MVNCD function from above is going to be at a low value or high value. But, the fact that the relative performances of the approaches do not change based on the context provides additional reinforcement to our findings that the TVBS approach appears to be most promising for all dimensions and all MVNCD evaluation contexts. The performance of the BME method is also very good, and may be considered in place of TVBS, especially for H=10 and beyond, if computational time is a critical issue.

4.2. MNP Model Results

Before presenting the MNP results, three important issues are in order. First, in examining the ability of the different evaluation methods to recover the true estimates, we tested a couple of implementation variants of the methods. The first uses the optimally-ordered abscissae approach to evaluate each of the individual choice probabilities, which, as indicated earlier, produced better individual MVNCD evaluations than without any such ordering. However, estimation of the underlying model parameters is not simply about estimating each individual choice probability accurately. As Bhat (2011) points out, "…. the focus in model estimation is on evaluating underlying model parameters that appear in the integrands of several multidimensional integrals.

The intent …is to estimate the model parameters accurately, and not expressly on evaluating each integral (that is, each individual choice probability) itself accurately." Indeed, this is the basis for the use of simulation methods that rely on the cancellation of simulation noise across individual choice probability evaluations. In our context, where we use analytic approximations rather than simulation methods, there may be value to randomizing the ordering of abscissae across individual observations. Doing so effectively randomizes the ordering of the deterministic utility differences as well as the covariance matrix of the utility differences, which may help cancel out "noise" introduced by the ordering effect. To test for this, our second variant in implementing the many analytic methods entailed randomizing the ordering of abscissae for each individual (with different randomizations for different individuals) rather than using a uniform optimal ordering across all individuals, as in the first variant (note, however, that the SSJ method is already based on random ordering, because its basis does not provide any advantage in terms of accuracy for an optimal ordering as done for the non-SSJ methods). Between the two variants -- optimally ordered abscissae approach and the randomly ordered abscissae approach -- the former came out to be the clear winner in terms of almost all the measures, suggesting that, even for parameter estimation, it is better in the context of analytic approximations to evaluate individual choice probabilities (and, thus, individual log-likelihoods) accurately rather than relying on any ordering noise cancellation through randomization of the abscissae across observations (the only exception to this was for the ME and the BME methods, which did see a marginal improvement in parameter recovery in the randomized abscissae variant, but this improvement was drowned by the degradation in the standard error recovery and individual choice probability evaluations).[9] In the rest of this section, we present the results only for the case where the abscissae for each individual are optimally ordered before computing the MVNCD function.

The second point is that, even if the randomization across individuals in the abscissae may help in potentially more accurately estimating the underlying model parameters, the asymptotic

[9] This general result held regardless of the starting values used for the MNP iterations. More generally, in another informal mini-experiment, we experimented with different starting values to test if the ability to recover the true estimates and the convergence times of the different evaluation methods is a function of the starting values. But the results for relative performance and computational time across the many methods tested remained stable regardless of starting point. Also, in our mini-experiments, the convergent estimates remained the same regardless of the starting point for each (and all) evaluation methods. In the subsequent reporting of computation time for convergence for the different evaluation methods, we provide the time for the specific case when the starting values are set at the common parameter values used to generate the many data samples (not the true estimates unique to each data sample).

standard errors of parameters are estimated based on the variation (across individuals) of the individual log-likelihood functions. Thus, in computing the standard errors, the accurate computation of each individual choice probability (and log-likelihood) at the estimated parameters should be of paramount importance. Similarly, while undertaking statistical tests or for predictions of the impacts of changes in exogenous variables, the ability to compute individual log-likelihoods and choice probabilities accurately is important. Thus, we also tested an implementation where, once parameters are estimated, the standard errors as well as all the likelihood-based measures are estimated by reverting to the optimal ordering of abscissae for each observation. In our experiments, however, we found very marginal improvements in the standard error recovery and the likelihood-based measures due to an optimal ordering after first estimating the parameters using a random ordering procedure.

Third, we confine attention in the MNP estimation to the analytic approximation methods, given that these methods are much faster than the traditional GHK simulation method for each individual MVNCD evaluation, while also being, in general, more accurate. Besides, the performance of the GHK degrades very rapidly as the dimensionality of the MVNCD function increases, as already discussed (see also Connors et al., 2014). In addition, Patil et al. (2017) have already established the superior performance of the SSJ1 method (the only analytic approximation method considered there) relative to the GHK method for the recovery of MNP choice model parameters (see also Bhat et al., 2010, Bhat and Sidharthan, 2011, Fu and Juan, 2017, and Martinetti and Geniaux, 2017, all of whom establish the superior performance of the SSJ1-based and/or ME-based methods relative to GHK-based methods for a whole variety of probit-kernel econometric models). Further, the TGME and TGBME methods performed very poorly relative to the other analytic approximation methods (by an order of magnitude), and so the results for these two methods are not presented in this section.[10]

Table 2 presents the results. In terms of the ability to recover the true estimates of the parameters (see first row panel of the table), for all methods except the ME, the mean absolute percentage error (MAPE) increases as the number of alternatives in the MNP model increases.

[10] We should note here that Martinetti and Geniaux (2017), in their spatial binary probit model, use the TGME method for evaluating the MVNCD function. They indicate reasonable performance of the method, though they did not compare the performance relative to the other methods discussed in this paper. On the other hand, Bhat and colleagues (see Sidharthan and Bhat, 2012, Castro et al., 2012, Ferdous and Bhat, 2013, Castro et al., 2013, Narayanamoorthy et al., 2013, Bhat et al., 2014, Bhat, 2015, Bhat et al., 2015, and Bhat et al., 2017) have used a MACML approach for estimating a variety of spatial models in which they use the SSJ approach for evaluating the MVNCD function.

This is to be expected. Among the many analytic methods, the SSJ1 method performs quite well, and there does not seem to be substantial gains from using the SSJ10 method (in fact, the SSJ10 method does provide MAPE values that are worse off than the faster SSJ1 method for the case with 11, 16, and 21 alternatives). Among the other methods, the ME and the BME methods are clearly not as good as the SSJ, OVUS, OVBS, and the TVBS methods in terms of parameter recovery, and the OVBS method appears to come out on top regardless of the number of alternatives involved. However, in terms of recovering the standard errors of the estimated parameters (second row panel of Table 2), the ME and BME do better than the other methods, while the TVBS method comes in third. On the dimensions of the ability to evaluate the overall log-likelihood function (third row panel), the ability to evaluate the individual likelihood function (that is, the choice probability of the chosen alternative) (the fourth row panel), and the ability to evaluate individual log-likelihoods (the fifth row panel), the OVBS and the TVBS generally come out well on top relative to other methods (except for the OVUS method, which comes in a close third). Interestingly, the SSJ1 method does the worst on these dimensions, though it does better than other methods except the OVBS method in terms of recovering the true parameter estimates. This suggests that it is not necessarily true that a method that recovers parameter estimates well will also approximate individual choice probabilities well. Finally, on the dimension of computation time, the SSJ10 method takes too long a time to be of practical value. The SSJ1 method, while quite fast for lower dimensions, tends to take rather long as the number of alternatives increases. The OVUS method appears to have a clear advantage over other methods in terms of the time to convergence, especially as the number of alternatives increase. Particularly important to note also is that the TVBS method is faster than the ME, OVBS, and BME methods by a rather substantial amount at $H+1 = 11$, 16, and 21. Overall, it appears that the OVUS, OVBS, or TVBS may be good candidates to consider.

To summarize, based on all the evidence thus far in terms of individual MVNCD function evaluations as well as MNP estimation, it appears that the OVUS, OVBS, and the TVBS methods are good candidates for consideration. The OVBS, in the context of MNP estimation takes rather long, while the OVUS method has a clear computational time advantage. The computational time for the TVBS method falls somewhere in-between. If we had to choose one uniform method for MVNCD function evaluation in any context, based on accuracy and computational time, we would favor the TVBS approach, based on the results in this paper.

5. CONCLUSIONS

In this paper, we develop new analytic ways to evaluate the multivariate normal cumulative distribution (MVNCD) function. These methods should be of interest in a wide variety of settings where the MVNCD function needs to be evaluated, including (but not limited to) the estimation of probit-based consumer choice models. Unlike traditional simulation-based methods to MVNCD evaluation for econometric models, which can be saddled with convergence and computational cost problems, these analytic approximation techniques for MVNCD evaluation are known to provide likelihood surfaces (and the derivatives and hessians of these surfaces with respect to model parameters) that are more smooth, reducing convergence and covariance matrix computation problems that can occur routinely in the maximum likelihood estimation of consumer choice models with analytically intractable likelihood functions (see Bhat and Sidharthan, 2011).

In the context of analytic approximation-based evaluations of MVNCD functions, while there have been many such approximations proposed, two of the most common are the SSJ and the ME approaches. In the current paper, we have proposed a streamlined and matrix-based version of the ME method, which relies on a single-sided truncation of a multivariate normal distribution in which some variables are truncated while others are not. But we propose a new way to implement the ME approach using an LDLT decomposition method followed by a rank 1 update of the LDLT factorization. Our implementation is easy to code for analysts skilled in matrix-based coding. Further, our new matrix-based implementation for the ME algorithm allows us to write, in a streamlined manner, the analytic matrix-based gradients of the approximated MVNCD function with respect to the abscissae and correlation parameters, an issue that is important in model estimation. In addition, we have proposed four new methods for approximating the MVNCD function, based on recognizing that, when untruncated variables are normally distributed, the marginal distribution of one of the untruncated variates given that other variables are truncated (or screened) is skew-normally distributed and not normally distributed. A rank-2 update of the LDLT decomposition is proposed and used in two of the newly proposed methods.

The paper evaluated the ability of the four proposed analytic MVNCD approximations, along with the SSJ, ME, and GHK simulation methods, for MVNCD function estimation for different numbers of dimensions of integration (H = 5, 7, 10, 12, 15, 18, and 20). In addition, the paper also evaluated the performance of the many analytic approximation methods in their ability

to recover underlying MNP parameters. For the latter evaluation, we use a key insight that allows us to obtain the true estimates of the underlying parameters accurately and quickly, given a data sample and regardless of the dimensionality involved in the resulting MVNCD evaluation.

As expected, in our tests for evaluating MVNCD functions, we found that the traditional GHK approach degrades rapidly as the dimensionality of integration increases. Concomitant with this degradation in accuracy is a rapid increase in computational time. The analytic approximation methods are more stable across different numbers of dimensions of integration, and even the simplest of these methods is superior to the GHK-500 beyond seven dimensions of integration. The more advanced analytic approximations proposed in this paper are superior to the GHK-500 at every integration dimension. Further, almost all of the methods provide better accuracy than even the GHK-10000 draws, and at about 12-14 times faster speed, at H=20. Also, when the truncation value from above is low, the GHK (and also the SSJ) approaches perform poorly relative to the other analytic approximation methods. Overall, we find that the TVBS approach is the best in terms of accuracy and speed in evaluating individual MVNCD functions.

In the testing of the analytic methods for MNP model estimation and prediction, we considered both an optimally ordered abscissae approach and a randomly ordered abscissae approach. The former came out to be the clear winner suggesting that, even for parameter estimation, it is better (at least in the context of analytic approximations) to evaluate individual choice probabilities (and, thus, individual log-likelihoods) accurately rather than relying on any ordering noise cancellation through randomization of the abscissa across observations. Overall, based on the multiple evaluation criteria of ability to recover parameters, ability to evaluate the overall log-likelihood function, ability to evaluate individual choice probabilities, ability to evaluate individual log-likelihoods, and computational time, the OVUS, OVBS, and the TVBS generally come out well on top relative to other methods.

Based on all the evaluation results in this paper, we recommend the TVBS approach as the one-stop evaluation approach for MVNCD function evaluation. Of course, further evaluation and testing of the many analytic approximation methods in other consumer choice modeling contexts that involve MVNCD function evaluations, such as in multivariate ordered-response models and mixed models with different types of dependent variables, would be helpful in additional assessments of the many methods proposed in the paper. Doing so is particularly important because the analytic methods, unlike the simulation methods, do not readily provide error estimates.

ACKNOWLEDGMENTS

This research was partially supported by the U.S. Department of Transportation through the Data-Supported Transportation Operations and Planning (D-STOP) Tier 1 University Transportation Center. The author would like to acknowledge support from a Humboldt Research Award from the Alexander von Humboldt Foundation, Germany. The author is grateful to Lisa Macias for her help in formatting this document, and to Sebastian Astroza for help with simulation runs and results preparation. Three anonymous reviewers provided useful comments on an earlier version of this paper.

REFERENCES

Aitken, A.C. (1934) Notes on selection from a multivariate normal population. *Proceedings of the Edinburg Mathematical Society*, 4(2), 106-110.

Arnold, B.C., Beaver, R.J., Groeneveld, R.A., and Meeker, W.Q. (1993) The nontruncated marginal of a truncated bivariate normal distribution. *Psychometrika*, 58(3), 471-488.

Batram, M., and Bauer, D. (2016) New results on the asymptotic and finite sample properties of the MaCML approach to multinomial probit model estimation. https://arxiv.org/abs/1609.03295.

Bhat, C.R. (2001) Quasi-random maximum simulated likelihood estimation of the mixed multinomial logit model. *Transportation Research Part B*, 35(7), 677-693.

Bhat, C.R. (2003) Simulation estimation of mixed discrete choice models using randomized and scrambled Halton sequences. *Transportation Research Part B*, 37(9), 837-855.

Bhat, C.R. (2011) The maximum approximate composite marginal likelihood (MACML) estimation of multinomial probit-based unordered response choice models. *Transportation Research Part B*, 45 (7), 923-939.

Bhat, C.R. (2015) A new spatial (social) interaction discrete choice model accommodating for unobserved effects due to endogenous network formation. *Transportation*, 42(5), 879-914.

Bhat, C.R., and Sidharthan, R. (2011) A simulation evaluation of the maximum approximate composite marginal likelihood (MACML) estimator for mixed multinomial probit models. *Transportation Research Part B*, 45(7), 940-953.

Bhat, C.R., Varin, C., and Ferdous, N. (2010) A comparison of the maximum simulated likelihood and composite marginal likelihood estimation approaches in the context of the multivariate ordered-response model. In Greene, W.H. and Hill, R.C. (Eds.) *Advances in Econometrics: Maximum Simulated Likelihood Methods and Applications*, 26, 65-106, Emerald Group Publishing Limited.

Bhat, C.R., Paleti, R., and Singh, P. (2014) A spatial multivariate count model for firm location decisions. *Journal of Regional Science*, 54(3), 462-502.

Bhat, C.R., Dubey, S.K., Jobair Bin Alam, M., and Khushefati, W.H. (2015) A new spatial multiple discrete-continuous modeling approach to land use change analysis. *Journal of Regional Science*, 55(5), 801-841.

Bhat, C.R., Astroza, S., and Hamdi, A. (2017) A spatial generalized ordered-response model with skew normal kernel error terms with an application to bicycling frequency. *Transportation Research Part B*, 95, 126-148.

Castro, M., Paleti, R., and Bhat, C.R. (2012) A latent variable representation of count data models to accommodate spatial and temporal dependence: application to predicting crash frequency at intersections. *Transportation Research Part B*, 46(1), 253-272.

Castro, M., Paleti, R., and Bhat, C.R. (2013) A spatial generalized ordered response model to examine highway crash injury severity. *Accident Analysis and Prevention*, 52, 188-203.

Clark, C. (1961) The greatest of a finite set of random variables. *Operations Research*, 9(2), 145-162.

Connors, R.D., Hess, S., and Daly, A. (2014) Analytic approximations for computing probit choice probabilities. *Transportmetrica A: Transport Science*, 10(2), 119-139.

Ferdous, N., and Bhat, C.R. (2013) A spatial panel ordered-response model with application to the analysis of urban land-use development intensity patterns. *Journal of Geographical Systems*, 15(1), 1-29.

Fu, X., and Juan, Z. (2017) Estimation of multinomial probit-kernel integrated choice and latent variable model: Comparison on one sequential and two simultaneous approaches. *Transportation*, 44(1), 91-116.

Geweke, J. (1991) Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints. In *Computer Science and Statistics: Proceedings of the Twenty Third Symposium on the Interface*, 571-578, Foundation of North America Inc., Fairfax Station, Va.

Gibson, G.J., Glasbey, C.A., and Elston, D.A. (1994) Monte Carlo evaluation of multivariate normal integrals and sensitivity to variate ordering. In Dimov, I.T., Sendov, B., Vassilevski, P.S. (eds.) *Advances in Numerical Methods and Applications*, pp. 120-126. World Scientific Publishing, River Edge, NJ.

Golub, G.H., and Van Loan, C.F. (2013) *Matrix Computations*, 4th Edition. The Johns Hopkins University Press, Baltimore.

Hajivassiliou, V., and McFadden, D. (1998) The method of simulated scores for the estimation of LDV models. *Econometrica*, 66(4), 863-896.

Heiss, F. (2010) The panel probit model: Adaptive integration on sparse grids. In Greene, W.H. and Hill, R.C. (Eds.) *Advances in Econometrics: Maximum Simulated Likelihood Methods and Applications*, Vol. 26, 41-64, Emerald Group Publishing Limited.

Heiss, F., and Winschel, V. (2008) Likelihood approximation by numerical integration on sparse grids. *Journal of Econometrics*, 144(1), 62-80.

Hutmacher, M.M., and French, J.L. (2011) Extending the latent variable model for extra correlated longitudinal dichotomous responses. *Journal of Pharmacokinetics and Pharmacodynamics*, 38(6), 833-859.

Imai, K., and van Dyk, D.A. (2005) A Bayesian analysis of the multinomial probit model using marginal data augmentation. *Journal of Econometrics*, 124(2), 311-334.

Joe, H. (1995) Approximations to multivariate normal rectangle probabilities based on conditional expectations. *Journal of the American Statistical Association*, 90(431), 957-964.

Kamakura, W.A. (1989) The estimation of multinomial probit models: A new calibration algorithm. *Transportation Science*, 23(4), 253-265.

Kan, R., and Robotti, C. (2017) On moments of folded and truncated multivariate normal distributions. *Journal of Computational and Graphical Statistics*, 26(4), 930-934.

Keane, M. (1990) Four essays in empirical macro and labor economics. PhD Thesis, Brown University.

Keane, M. (1994) A computationally practical simulation estimator for panel data. *Econometrica*, 62(1), 95-116.

Kim, H-J., and Kim, H-M. (2015) A class of rectangle-screened multivariate normal distributions and its applications. *Statistics*, 49(4), 878-899.

Kotz, S., Balakrishnan, N., and Johnson, N.L. (2000) *Continuous Multivariate Distributions, Volume 1: Models and Applications*, 2nd Edition, John Wiley & Sons, Inc., New York.

Langdon, M.G. (1984) Improved algorithms for estimation choice probabilities in the multinomial probit model. *Transportation Science*, 18(3), 267-299.

Lee, S.X., and McLachlan, G.J. (2013) On mixtures of skew-normal and skew t-distributions. *Advances in Data Analysis and Classification*, 7, 241-266.

Manjunath, B.G., and Wilhelm, S. (2012) Moments calculation for the doubly truncated multivariate normal density. https://arxiv.org/abs/1206.5387.

Martinetti, D., and Geniaux, G. (2017) Approximate likelihood estimation of spatial probit models. *Regional Science and Urban Economics*, 64, 30-45.

McCulloch, R.R., Polson, N.G., and Rossi, P.E. (2000) Bayesian analysis of the multinomial probit model with fully identified parameters. *Journal of Econometrics*, 99, 173-193.

Mendell, N.R., and Elston, R.C. (1974) Multifactorial qualitative traits: Genetic analysis and prediction of recurrence risks. *Biometrics*, 30(1), 41-57.

Narayanamoorthy, S., Paleti, R., and Bhat, C.R. (2013) On accommodating spatial dependence in bicycle and pedestrian injury counts by severity level. *Transportation Research Part B*, 55, 245-264.

Patil, P.N., Dubey, S.K., Pinjari, A.R., Cherchi, E., Daziano, R. and Bhat, C.R. (2017) Simulation evaluation of emerging estimation techniques for multinomial probit models. *Journal of Choice Modelling*, 23, 9-20.

Pearson, K. (1903) On the influence of natural selection on the variability and correlation of organs. *Philosophical Transactions of the Royal Society*, A200, 1-66.

Rice, J., Reich, T., Cloninger, C.R., and Wette, R. (1979) An approximation to the multivariate normal integral: Its application to multifactorial qualitative traits. *Biometrics*, 35(2), 451-459.

Sidharthan, R., and Bhat, C.R. (2012) Incorporating spatial dynamics and temporal dependency in land use change models. *Geographical Analysis*, 44(4), 321-349.

Solow, A.R. (1990) A method for approximating multivariate normal orthant probabilities. *Journal of Statistical Computation and Simulation*, 37(3-4), 225-229.

Switzer, P. (1977) Estimation of spatial distributions from point sources with application to air pollution measurement. *Bulletin of the International Statistical Institute*, 47, 123-137.

Trinh, G., and Genz, A. (2015) Bivariate conditioning approximations for multivariate normal probabilities. *Statistics and Computing*, 25(5), 989-996.

Table 1. MVNCD evaluation results for the different analytic approximation methods

Dim.	Measure	GHK		SSJ		Univariate truncation methods				Bivariate truncation methods		
		500	10,000	1	10	TGME	ME	OVUS	OVBS	TGBME	BME	TVBS
	MAE	0.00061	0.00009	0.00182	0.00116	0.01514	0.00124	0.00078	0.00045	0.01065	0.00083	0.00051
	MAPE	0.92	0.14	3.77	2.90	9.08	1.78	1.52	0.98	6.72	1.32	0.82
H=5	%MAE>0.005	0.9	0.2	9.0	3.8	56.0	6.3	2.9	1.0	47.3	2.7	1.1
	%MAPE>2	4.3	0.9	27.6	19.8	65.0	13.4	9.3	5.1	54.5	8.0	4.9
	Time (s)	0.071	0.989	0.035	0.222	0.031	0.063	0.073	0.072	0.037	0.050	0.066
	MAE	0.00075	0.00009	0.00148	0.00085	0.01301	0.00081	0.00064	0.00043	0.01020	0.00061	0.00045
	MAPE	1.77	0.43	7.44	5.36	11.34	1.94	1.83	1.83	8.77	1.62	1.46
H=7	%MAE>0.005	1.4	0.1	6.7	1.5	53.9	3.1	1.8	0.9	48.4	1.7	0.9
	%MAPE>2	11.4	2.2	38.1	28.9	83.6	14.3	11.9	8.9	75.8	10.2	7.9
	Time (s)	0.106	1.670	0.054	0.389	0.048	0.114	0.126	0.141	0.055	0.078	0.122
	MAE	0.00096	0.00011	0.00102	0.00057	0.01191	0.00050	0.00042	0.00032	0.00976	0.00040	0.00032
	MAPE	5.30	1.93	10.23	8.31	14.78	2.83	2.85	2.71	12.81	3.16	3.21
H=10	%MAE>0.005	3.6	0.0	2.7	0.6	47.8	1.1	0.8	0.5	45.7	0.6	0.3
	%MAPE>2	26.1	8.6	45.3	38.3	89.7	16.7	15.2	12.6	87.9	13.9	12.7
	Time (s)	0.172	3.029	0.114	0.791	0.067	0.213	0.233	0.270	0.080	0.126	0.210
	MAE	0.00101	0.00012	0.00085	0.00049	0.01078	0.00038	0.00032	0.00025	0.00898	0.00031	0.00025
	MAPE	7.00	2.16	15.66	12.06	14.62	3.29	2.97	2.71	12.52	2.68	2.41
H=12	%MAE>0.005	4.5	0.0	2.5	0.4	46.8	0.2	0.1	0.0	44.1	0.1	0.1
	%MAPE>2	35.5	10.9	47.7	42.4	92.7	18.3	17.0	14.6	91.4	15.1	13.5
	Time (s)	0.219	4.653	0.150	1.145	0.084	0.296	0.328	0.383	0.095	0.187	0.295
	MAE	0.00111	0.00014	0.00077	0.00038	0.00927	0.00029	0.00026	0.00020	0.00792	0.00024	0.00020
	MAPE	10.86	4.83	14.53	12.23	16.21	4.13	3.76	3.68	14.62	3.68	3.71
H=15	%MAE>0.005	4.3	0.0	1.4	0.0	45.4	0.2	0.1	0.1	45.0	0.1	0.1
	%MAPE>2	47.1	18.2	51.5	43.7	93.3	21.3	20.2	17.9	93.1	17.7	16.7
	Time (s)	0.291	8.818	0.268	1.757	0.118	0.559	0.570	0.584	0.135	0.265	0.422
	MAE	0.00113	0.00015	0.00059	0.00034	0.00855	0.00024	0.00021	0.00016	0.00737	0.00019	0.00016
	MAPE	15.55	6.96	16.67	13.33	17.92	5.41	4.83	4.56	15.95	4.06	3.67
H=18	%MAE>0.005	5.4	0.0	0.3	0.0	43.8	0.0	0.0	0.0	42.3	0.0	0.0
	%MAPE>2	56.7	28.0	52.7	45.4	93.3	24.6	23.6	23.1	93.8	20.9	20.4
	Time (s)	0.400	9.252	0.394	2.653	0.137	0.675	0.694	0.843	0.166	0.368	0.571
	MAE	0.00124	0.00016	0.00057	0.00032	0.00789	0.00021	0.00018	0.00015	0.00681	0.00017	0.00015
	MAPE	18.53	10.76	17.22	14.69	20.07	5.37	5.10	5.11	17.89	4.87	5.08
H=20	%MAE>0.005	6.9	0.0	0.8	0.0	40.2	0.0	0.0	0.0	39.9	0.0	0.0
	%MAPE>2	63.3	34.1	53.7	47.3	94.7	26.0	24.9	24.7	93.9	23.8	24.0
	Time (s)	0.458	11.390	0.426	3.093	0.155	0.800	0.879	0.987	0.197	0.437	0.652

Table 2. MNP estimation results for the different analytic approximation methods

Number of alternatives	Analytics Approximation Methods						
	SSJ		Univariate truncation methods			Bivariate truncation methods	
	1	10	ME	OVUS	OVBS	BME	TVBS
Ability to recover "true" parameter estimates (absolute percentage error)							
6	1.62	1.34	9.09	3.39	1.34	6.04	1.95
11	3.19	3.27	9.01	3.92	2.65	7.92	4.24
16	4.66	4.86	8.37	4.76	3.31	8.27	5.02
21	4.53	5.52	8.08	4.95	3.77	8.79	6.00
All	3.50	3.75	8.64	4.25	2.77	7.75	4.30
Ability to recover "true" sampling standard error of estimator (absolute percentage error)							
6	12.89	12.84	9.19	11.68	12.67	10.39	12.31
11	11.01	10.92	7.93	10.02	10.67	8.10	9.77
16	8.64	8.33	6.38	8.01	8.71	6.20	7.53
21	8.75	7.52	5.99	7.14	7.62	5.42	6.46
All	10.32	9.90	7.38	9.21	9.92	7.53	9.02
Ability to estimate overall log-likelihood function (absolute percentage error)							
6	0.11	0.09	0.40	0.06	0.04	0.56	0.09
11	0.45	0.27	0.16	0.28	0.23	0.56	0.06
16	0.81	0.48	0.10	0.49	0.42	0.37	0.09
21	1.07	0.56	0.40	0.72	0.62	0.11	0.29
All	0.61	0.35	0.27	0.39	0.33	0.40	0.14
Ability to replicate the individual likelihood function values (absolute percentage error)							
6	2.60	1.14	3.03	1.04	0.63	2.36	0.80
11	5.85	2.49	3.80	1.78	1.25	3.76	1.81
16	7.95	3.52	3.93	2.44	1.93	4.22	2.46
21	10.39	4.65	4.14	3.20	2.71	4.74	3.11
All	6.70	2.95	3.73	2.11	1.63	3.77	2.04
Ability to replicate the logarithm of the individual likelihood function (absolute percentage error)							
6	1.97	0.89	2.68	0.94	0.56	2.04	0.71
11	3.49	1.49	2.75	1.27	0.86	2.69	1.28
16	4.27	1.98	2.59	1.49	1.15	2.77	1.56
21	4.94	2.34	2.35	1.64	1.38	2.75	1.71
All	3.67	1.68	2.59	1.33	0.99	2.56	1.31
Computation time (minutes)							
6	0.69	4.36	1.13	1.20	2.89	1.21	2.69
11	7.12	57.96	12.81	6.77	12.31	20.60	9.05
16	32.08	259.20	24.33	23.13	30.35	42.76	22.19
21	221.80	2366.67	180.13	63.40	195.99	189.84	127.58
All	65.42	672.05	54.60	23.62	60.38	63.60	40.38

Appendix A: The covariance matrix of single-sided truncations (from above) of bivariate normally distributed random variables

Using the same notations as in Property 1, and using the general results on the moments of truncated multivariate distributions (see Kan and Robotti, 2017) we may write the following:

$$E(X_1^2 \mid X_1 < x_1, X_2 < x_2) =$$

$$\sigma_1^2\left[1+\left(\frac{\mu_1}{\sigma_1}\right)^2 - \frac{1}{\Phi_2(w_1,w_2,\rho)}\left\{\left(\frac{\mu_1+x_1}{\sigma_1}\right)\delta_1 + \rho\left[\left(1-\rho^2\right)\phi(w_1,w_2,\rho) - \left(\frac{2\mu_1}{\sigma_1}+\rho w_2\right)\delta_2\right]\right\}\right]$$

$$E(X_{21}^2 \mid X_1 < x_1, X_2 < x_2) =$$

$$\sigma_2^2\left[1+\left(\frac{\mu_2}{\sigma_2}\right)^2 - \frac{1}{\Phi_2(w_1,w_2,\rho)}\left\{\left(\frac{\mu_2+x_2}{\sigma_2}\right)\delta_2 + \rho\left[\left(1-\rho^2\right)\phi(w_1,w_2,\rho) - \left(\frac{2\mu_2}{\sigma_2}+\rho w_1\right)\delta_1\right]\right\}\right]$$

$$E(X_1X_2 \mid X_1 < x_1, X_2 < x_2) =$$

$$\mu_1\mu_2 + \sigma_1\sigma_2\left[\rho - \frac{1}{\Phi_2(w_1,w_2,\rho)}\left\{\delta_1\left(\frac{\mu_2}{\sigma_2}+\rho w_1+\frac{\rho\mu_1}{\sigma_1}\right) + \delta_2\left(\frac{\mu_1}{\sigma_1}+\rho w_2+\frac{\rho\mu_2}{\sigma_2}\right) - \left(1-\rho^2\right)\phi_2(w_1,w_2,\rho)\right\}\right]$$

$$Var(X_1^2 \mid X_1 < x_1, X_2 < x_2) = \ddot{\sigma}_1^{\,2} = E(X_1^2 \mid X_1 < x_1, X_2 < x_2) - \left[E(X_1 \mid (X_1 < x_1, X_2 < x_2))\right]^2$$

$$Var(X_2^2 \mid X_1 < x_1, X_2 < x_2) = \ddot{\sigma}_2^{\,2} = E(X_2^2 \mid X_1 < x_1, X_2 < x_2) - \left[E(X_2 \mid (X_1 < x_1, X_2 < x_2))\right]^2$$

$$Cov(X_1X_2 \mid X_1 < x_1, X_2 < x_2) = \ddot{\sigma}_{12} = E(X_1X_2 \mid X_1 < x_1, X_2 < x_2) - \left[E(X_1 \mid (X_1 < x_1, X_2 < x_2))E(X_2 \mid (X_1 < x_1X_2 < x_2))\right]$$

Using the expressions above and the expressions for $E(X_1 \mid (X_1 < x_1, X_2 < x_2))$ and $E(X_2 \mid (X_1 < x_1, X_2 < x_2))$ from Equation (3), and after some straightforward but tedious algebra, the simple expressions in Equation (5) result.