Initialization for NORTA: Generation of Random Vectors with Specified Marginals and Correlations

Huifen Chen

Department of Industrial Engineering, Chung Yuan Christian University, Chung Li, 320, Taiwan

huifen@cycu.edu.tw

We propose a specific method for generating n-dimensional random vectors with given marginal distributions and correlation matrix. The method uses the NORTA (NORmal To Anything) approach, which generates a standard normal random vector and then transforms it into a random vector with specified marginal distributions. During initialization, $n(n-1)/2$ nonlinear equations need to be solved to ensure that the generated random vector has the specified correlation structure. To solve these equations, we apply retrospective approximation, a generic stochastic root-finding algorithm, with slight changes. Internal control variates are used to estimate function values. Empirical comparisons show that the control-variate variance-reduction technique improves the algorithm's convergence speed as well as its robustness. Simulation results for a variety of marginal distributions and correlation matrices are also presented.

(*Control Variate; Retrospective Approximation; Root Finding; Simulation*)

---

1. Introduction

We consider the problem of generating observations of the random vector $\underline{X} = (X_1, ..., X_n)$ with correlation matrix $(\rho_{ij})_{i,j=1}^n$ and marginal distributions $F_1, ..., F_n$. In simulation experiments, input models may include a set of correlated random variables, e.g., market size and selling price in capital-return models (Hull 1977), and correlated attributes in decision analysis (Corner and Kirkwood 1996). To perform such simulation experiments, efficient methods for generating correlated random variates are needed.

The literature on generation of correlated random variates falls into two categories (Nelson and Yamnitsky 1998, Schmeiser 1999): (1) time-series input processes and (2) random vectors. For time-series input processes, Gaver and Lewis (1980) propose an algorithm to

generate gamma time-series data, whose autocorrelation is of geometric functional form. Gujar and Kavanagh (1968), Broste (1971), Song and Hsiao (1993), Song et al. (1996), and Cario and Nelson (1996, 1998) propose algorithms for stationary time-series data with an arbitrary marginal distribution and an arbitrary autocorrelation structure.

For random-vector generation (RVG), the literature focuses mainly on random vectors with marginal distributions from the same standard family. For example, Moonan (1957) proposes the generation of normal random vectors based on the linear transformation of a set of independent standard-normal random variables; Ronning (1977), Schmeiser and Lal (1982), and Lewis (1983) propose generation methods for the nonnegatively correlated multivariate gamma, bivariate gamma, and negatively correlated multivariate gamma distributions, respectively. Other types of multivariate distributions in the literature include the multivariate lognormal (Johnson and Ramberg 1978), multivariate Pearson (Parrish 1990), and multivariate Johnson (Johnson 1987, Stanfield et al. 1996) distributions. We discuss generic RVG methods in Section 2.1.

We are interested in RVG methods that allow different, discrete, and non-standard marginal distributions, defined through a user-provided computer routine. We require that such a routine generate variates from each marginal distribution using the inverse-transformation method. Specifically, our research problem is defined as follows.

Random-Vector Generation Problem:

Given:

(a) the dimension n of the random vector $\underline{X}$,

(b) a computer routine to compute F_1^{-1},...,F_n^{-1}, the inverse functions of F_1,...,F_n, and

(c) a feasible correlation matrix $\Sigma_X = (\rho_{ij})_{i,j=1}^n$.

Develop: a simulation algorithm to generate observations of the random vector $\underline{X} = (X_1, ..., X_n)$ with correlation matrix Σ_X and marginal distributions $F_1, ..., F_n$.

The specified correlation matrix Σ_X must be feasible. Being nonnegative definite is a necessary but not sufficient condition for feasibility (Cario and Nelson 1996).

The RVG method that we use for this research problem is NORTA (NORmal To Anything, described in Section 2.2). The NORTA approach transforms a standard-normal random vector $\underline{Z}$ into the desired random vector $\underline{X}$. For each application, NORTA initialization

requires solving $n(n-1)/2$ one-dimensional equations for finding the value of $\underline{Z}$'s correlation matrix so that the generated $\underline{X}$ has the desired correlation matrix Σ_X. For this stochastic root-finding problem (SRFP, Chen and Schmeiser 1994a), we implement the retrospective approximation (RA) algorithm, by Chen and Schmeiser (1994b, 2001), with slight changes for increased efficiency. An earlier version of our RVG method (Chen and Jeng 1995) is implemented in Corner and Kirkwood (1996) to generate correlated attributes with gamma marginal distributions.

The rest of this paper is organized as follows. In Section 2, we discuss generic methods for RVG and review the NORTA approach and RA algorithm, which are then combined as a new RVG method as proposed in Section 3. During root finding, internal control variates are used to reduce the variance of function-value estimates. In Section 4, we study the variance reduction achieved by control variates and evaluate our RVG method based on simulation experiments with different marginal distributions and correlations. The results show that control variates significantly reduce the generalized mean square error (mse) and that the root-finding efficiency depends on the types of marginal distributions. Section 5 gives our conclusions.

2. Literature Review

2.1 Generic Methods for RVG

There are three generic approaches for generating multivariate random vectors: (1) conditional distribution, (2) acceptance/rejection, and (3) partially-specified-property transformation. The procedure for the first approach is: (i) generate X_1 from the marginal distribution F_1, and (ii) given $X_1, ..., X_{k-1}$, generate X_k from the conditional distribution $F(X_k|X_1, ..., X_{k-1})$ for $k = 2, ..., n$. Despite its simple logic, this approach may be difficult to apply because conditional distributions are not easy to derive except for special cases, e.g., multivariate normal distributions. Another drawback of this approach is that the joint distribution of the random vector is assumed known, whereas fitting a joint distribution to data may be difficult in practice.

The acceptance/rejection approach can be used for multi-dimensional random-variate generation (Devroye 1986, Johnson 1987). To generate random vectors from a multivariate distribution with joint density function $f(\underline{x})$, acceptance/rejection works as follows: A function $ch(\underline{x})$ that dominates $f(\underline{x})$ (i.e., $ch(\underline{x}) \geq f(\underline{x})$ for any $\underline{x}$ in the support D of f)

is selected and then a random vector $\underline{x}$ generated from the joint density function $h(\underline{x})$ is accepted with probability $f(\underline{x}) \ / \ [ch(\underline{x})]$. As with the conditional-distribution approach, the major drawback is that the joint density $f(\underline{x})$ needs to be known. Two application issues of acceptance/rejection are finding an efficient h from which it is easy to generate observations, and computing the constant $c = \sup_{\underline{x} \in D} f(\underline{x}) \ / \ h(\underline{x})$. If D is finite, a quick choice of h is the multivariate uniform; if D is infinite, a reasonable choice is to let h be the joint distribution of independent random variables with the same marginals as those of f. However, for these choices of h, the constant c (and hence the generation time) increases rapidly as the dimension increases. A more complicated h might be more efficient, but constructing h, generating from h, and computing c may be more difficult and expensive. Gilks and Wild (1992), Hörmann (1995), and Leydold (1998) suggest a transformed density rejection method to construct a dominating function, in which one uses a monotonic function to transform f into a concave function, takes the minimum of several tangent hyperplanes, and then transforms it back into the original scale.

The third approach transforms a multivariate random vector with a given set of marginals into a random vector with target marginals and correlation matrix (Law and Kelton 2000, p. 482). Unlike the first two approaches, the third approach assumes that only the marginal distributions and correlation matrix of $\underline{X}$ are known rather than the whole joint distribution. Input modeling under this assumption is easy. However, the drawback of this approach is that the joint distribution with the specified marginals and correlations is not unique. A specific approach in this family is NORTA—also called the marginal-oriented approach by Song and Hsiao (1993). NORTA generates a standard-normal random vector, transforms it into a vector of uniform random numbers, and then transforms the uniform random vector into the random vector with target marginals by the inverse-transformation method. We discuss the details of NORTA in Section 2.2. Another specific approach is the copula method that transforms a uniform random vector directly into the random vector of interest (Clemen et al. 2000, Clemen and Reilly 1999). This approach bypasses the first step in NORTA, but a copula must be created to ensure that the generated $\underline{X}$ has desired correlations. Clemen and Reilly (1999) model the dependence among uniform random variables by the rank-order correlations such as Spearman's or Kendall's correlation, rather than the Pearson correlations used in this paper, because the rank-order correlations do not depend on the type of marginal distributions.

2.2 NORTA for RVG

The NORTA approach can be used to generate observations of the random vector $\underline{X} = (X_1, ..., X_n)$ with given marginal cumulative distribution functions (cdfs) F_1,...,F_n and correlation matrix $(\rho_{ij})_{i,j=1}^n$. Earlier references include Mardia (1970, p. 30), Li and Hammond (1975), and Schmeiser (1990). We adopt the name NORTA from Cario and Nelson (1997). This approach is implemented as follows:

1. Generate a multivariate standard-normal random vector $\underline{Z} = (Z_1, ..., Z_n)$ such that $\text{Corr}(Z_i, Z_j) = \rho_{ij}^*$ for $1 \le i, j \le n$.
2. Compute $U_i = \Phi(Z_i)$ for $i = 1, ..., n$, where $\Phi(\cdot)$ is the standard normal cdf.
3. Compute $X_i = F_i^{-1}(U_i)$ for $i = 1, ..., n$, where F_i^{-1} is the inverse of F_i.

Steps 1 and 2 together generate a uniform random vector $(U_1, ..., U_n)$. Step 3 then uses the inverse-transformation method to transform the uniform random vector into $(X_1, ..., X_n)$ whose marginal distributions are $F_1, ..., F_n$. The standard-normal correlations ρ_{ij}^* in step 1 are chosen to ensure the desired correlations ρ_{ij} of the generated $(X_1, ..., X_n)$ in step 3.

Two implementation problems arise in step 1: (i) how to generate observations of the standard-normal random vector with correlations ρ_{ij}^*, $1 \le i, j \le n$, and (ii) how to specify the values of ρ_{ij}^*. The first problem can be solved by implementing the conditional-distribution approach or, preferably in general, by linearly transforming a set of n independent standard-normal random variates (Moonan 1957, Scheuer and Stoller 1962, Ripley 1987, p. 98, Law and Kelton 2000, p. 480). Suppose that random vector $\underline{V}$ contains n independent standard-normal random variables and C is the Cholesky decomposition of the correlation matrix $\Sigma_Z = (\rho_{ij}^*)_{i,j=1}^n$, i.e., $C^{\mathrm{T}}C = \Sigma_Z$. Then, $C^{\mathrm{T}}\underline{V}$ has a multivariate standard-normal distribution with correlation matrix Σ_Z.

The second problem, finding the value of the step-1 correlations ρ_{ij}^* so that the generated random variates X_i and X_j in step 3 have correlations ρ_{ij} for $1 \le i, j \le n$, is the focus of this research. Based on the nonlinear transformation from step 1 to step 3, each pair (ρ_{ij}^*, ρ_{ij}) satisfies Equation (1):

$$\begin{aligned} \rho_{ij} &= \text{Corr}(X_i, X_j) \\ &= \text{Corr}[F_i^{-1}(\Phi(Z_i)), F_j^{-1}(\Phi(Z_j))], \quad \text{because } \Phi(Z_i) = U_i = F_i(X_i), \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\frac{F_i^{-1}(\Phi(z_i)) - \mu_i}{\sigma_i}\right]\left[\frac{F_j^{-1}(\Phi(z_j)) - \mu_j}{\sigma_j}\right]\phi(z_i, z_j; \rho_{ij}^*)\, dz_i\, dz_j\ , \end{aligned} \tag{1}$$

where μ_i and σ_i^2 are the mean and variance respectively of the marginal distribution F_i, and ϕ is the bivariate standard-normal density function with correlation ρ_{ij}^*. Notice that

each ρ_{ij} functionally depends only on ρ^*_{ij}, not on any other step-1 correlations ρ^*_{pl}, where $(p,l) \neq (i,j)$. That is, Equation (1) is one dimensional.

Therefore, the second problem is equivalent to solving $n(n-1)/2$ one-dimensional equations of form (1) for ρ^*_{ij}, $1 \leq i < j \leq n$. (Notice that we need solve only $n(n-1)/2$, not n^2, equations because $\rho^*_{ij} = \rho^*_{ji}$ for $1 \leq i \neq j \leq n$ and $\rho^*_{ii} = 1$ for $1 \leq i \leq n$.) For convenience, we rewrite Equation (1) as $g_{ij}(\rho^*_{ij}) = \rho_{ij}$, where the root-finding function

$$g_{ij}(r) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\frac{F_i^{-1}(\Phi(z_i)) - \mu_i}{\sigma_i} \right] \left[\frac{F_j^{-1}(\Phi(z_j)) - \mu_j}{\sigma_j} \right] \phi(z_i, z_j; r)\, dz_i\, dz_j \;, \quad |r| \leq 1, \quad (2)$$

is the correlation between the X_i and X_j that are generated by NORTA using r as the correlation of Z_i and Z_j in step 1. Cario and Nelson (1997) have pointed out three important properties of the function $g_{ij}(r)$—stated in Theorems 1 and 2 and Proposition 1 below—that are useful for solving Equation (1).

Theorem 1 *The function $g_{ij}(r)$ is nondecreasing for $-1 \leq r \leq 1$.*

Theorem 2 *If there exists $\zeta > 0$ such that $\mathrm{E}[|X_i X_j|^{1+\zeta}] < \infty$ for all values of $-1 \leq r \leq 1$, where X_i, X_j are defined by a NORTA transformation, then the function $g_{ij}(r)$ is continuous for $-1 \leq r \leq 1$.*

Proposition 1 *Let $\bar{\rho}_{ij}$ and $\underline{\rho}_{ij}$ be the maximum and minimum feasible bivariate correlations, respectively, for random variables having marginal distributions F_i and F_j via a NORTA transformation. Then, $g_{ij}(1) = \bar{\rho}_{ij}$ and $g_{ij}(-1) = \underline{\rho}_{ij}$.*

Proofs can be found in Cario and Nelson (1997).

From Theorems 1 and 2 and the Intermediate Value Theorem, the solution ρ^*_{ij} of Equation (1) exists if the condition of Theorem 2 is satisfied and the specified step-3 correlation ρ_{ij} is in the interval $[\underline{\rho}_{ij}, \bar{\rho}_{ij}]$. However, even when each of the $n(n-1)/2$ equations $g_{ij}(\rho^*_{ij}) = \rho_{ij}$, $1 \leq i < j \leq n$, has a unique solution ρ^*_{ij}, the step-1 correlation matrix $(\rho^*_{ij})_{i,j=1}^n$ may be infeasible. We discuss the properties of g further in Section 3.

There are three general approaches for solving $n(n-1)/2$ equations of form (1): analytical, numerical and simulation. The analytical approach works for some special cases such as uniform random vectors (see Section 4). The numerical approach uses numerical root-finding methods to solve the $n(n-1)/2$ equations, where double-integral function values of form (2) are evaluated by numerical integration methods. The numerical integration methods include methods using equispaced abscissas, such as the trapezoid rule and Simpson's rule, and

methods using non-equispaced abscissas, such as Gaussian quadrature methods (Conte and de Boor 1980). Given a value of r, either type of method computes approximations of the function value $g_{ij}(r)$ by the form

$$g_{ij}(r) \approx \sum_{l=1}^{d}\sum_{k=1}^{d} w_l w_k \left[\frac{F_i^{-1}(\Phi(z_l)) - \mu_i}{\sigma_i}\right]\left[\frac{F_j^{-1}(\Phi(z_k)) - \mu_j}{\sigma_j}\right]\phi(z_l, z_k; r) \ , \qquad (3)$$

where z_1,..., z_d are d abscissas chosen on the axes of both Z_i and Z_j and the w's are weights. The methods with equispaced abscissas guarantee exact solutions for polynomials of degree $\leq d$, the number of abscissas evaluated; Gaussian quadrature methods guarantee exact solutions for polynomials of degree $\leq 2d-1$. Since all $n(n-1)/2$ equations are functionally independent, they can be solved separately by one-dimensional root-finding methods, such as regula falsi and Newton's methods. (The monotonicity of the functions g_{ij} is helpful for finding an initial bounding interval, which brackets the root, and deciding the direction of movements.) The computational time, however, increases quadratically with n. All equations can also be solved simultaneously and efficiently using the search procedure in the ARTAFACTS software (Cario and Nelson 1998). Notice that only $\phi(z_l, z_k; r)$ in the double sum depends on r. Therefore, given abscissas z_1,..., z_d, we can compute the values of $w_l[F_i^{-1}(\Phi(z_l)) - \mu_i]/\sigma_i$ for $l = 1, .., d$ and $i = 1, ..., n$ and store them prior to the root search. Then during the root search, we can evaluate $g_{12}(r_{12})$,..., $g_{n-1,n}(r_{n-1,n})$ simultaneously for any set $(r_{12}, ..., r_{n-1,n})$ of root candidates by computing only the $\phi(\cdot)$'s and summing up all terms in Equation (3). The total computation time consists of the time to compute $w_l[F_i^{-1}(\Phi(z_l)) - \mu_i]/\sigma_i$, which increases with n, and the time to compute the double sum, which increases with n^2. Unless n is very large, the first part dominates since computing F_i^{-1} is usually more time consuming.

When the integrands are smooth, the numerical approach is very efficient. Otherwise, equispaced integration methods may be inefficient and Gaussian quadrature methods may be inaccurate. Particularly when the step-1 correlation r is close to 1 (or -1), the bivariate normal density is high in the neighborhood of the 45-degree (or 135-degree) line and almost zero elsewhere. For example, when $n = 2$ and both F_1 and F_2 are U(0,1) distributions, the Gaussian Hermit approximation (with 32 abscissas) of $g_{12}(0.98)$ is 0.993. The true value of $g_{12}(0.98)$, however, is $(6/\pi)\sin^{-1}(0.98/2) = 0.978$ (see Section 4). Worse, when $r > 0.98$, the Gaussian Hermit approximation is larger than 1 . One remedy to overcome this difficulty is the composite rule, dividing the bivariate normal sample space into several subsquares, computing the integral on each subsquare, and then summing the integrals.

Analogously, the simulation approach treats solving the $n(n-1)/2$ equations as an SRFP by solving them simultaneously using estimates of the function values. For any set $(r_{12}, ..., r_{n-1,n})$ of root candidates, the function values $(g_{12}(r_{12}), ..., g_{n-1,n}(r_{n-1,n}))$ are correlations and hence can be estimated by the sample correlations from m random vectors that are generated via the NORTA transformation (e.g., Equation 6 or 7). The total computation time consists of the time required to generate observations and the time to compute the correlation estimates. The generation time has order $O(n)$ because all equations are solved simultaneously with m observations of dimension n (see Section 3). The time required to compute the $n(n-1)/2$ correlation estimates has order $O(n^2)$. Though the latter is of a higher order, usually the generation task consumes most of the computation effort unless n is very large. The major drawback of this approach is that during the root search, Cholesky decomposition in NORTA step 1 may fail. In this case, some equations have to be solved separately, instead of simultaneously, and hence the computation time increases (Section 3).

NORTA has been previously used for stationary time-series processes. Given a marginal distribution and the first p autocorrelations, Song and Hsiao (1993) and Song et al. (1996) solve Equation (1) stochastically, and Cario and Nelson (1996, 1998) solve it numerically, to generate time-series random variables. The software ARTAFACTS developed by Cario and Nelson (1998) solves all equations simultaneously and hence is very efficient.

In the RVG literature, Hull (1977) uses NORTA for RVG but uses the conditional distributions of $X_i|X_j$'s (assumed known) to solve Equation (1). Li and Hammond (1975) and Cario and Nelson (1997) solve Equation (1) numerically using Newton's method and the bisection method, respectively. However, they only report computational results, omitting the search procedures and computation times. Lurie and Goldberg (1998) propose an algorithm for generating observations of random vectors with continuous and strictly increasing marginal distributions. The algorithm works as follows: (i) Check whether the specified correlation matrix $(\rho_{ij})_{i,j=1}^n$ is nonnegative definite and adjust the matrix if it is not; (ii) Solve $n(n-1)/2$ equations of form (1) simultaneously by treating the root finding as a stochastic optimization problem, where the objective function is the squared distance between the candidate correlation matrix $(g_{ij}(r_{ij}))_{i,j=1}^n$ and the target correlation matrix $(\rho_{ij})_{i,j=1}^n$ of $\underline{X}$. Like our algorithm, it uses the simulation approach. Unlike ours, it solves the stochastic optimization problem using a single sample path with a big sample size and their correlation estimate is a crude estimate as defined in Equation (6). Ghosh and Henderson (2000) make a simple modification to the NORTA initialization phase that enables close approximation of the target

correlation matrix when the solved step-1 correlation matrix $(\rho^*_{ij})_{i,j=1}^n$ is not feasible. The modification involves solving a semidefinite program and works in both the rank correlation and Pearson product-moment correlation cases. Ghosh and Henderson also propose an LP method that can exactly match correlations not achievable via NORTA. However, the LP method can involve a nontrivial amount of initialization computation time.

We propose here an explicit Monte Carlo algorithm to estimate the step-1 correlations ρ^*_{ij} where the marginal distributions can be discrete. Viewing the root finding as an SRFP, we solve the equations by solving a sequence of sample-path equations with increasing sample sizes and decreasing error tolerances (Section 2.3). Internal control variates are used to improve the convergence speed. Figures 1 through 4 in Section 4 show the advantages of using the control variate and of solving a sequence of sample-path equations with a small initial sample size. In this research we assume that the NORTA step-1 correlation matrix is positive definite so that the Cholesky decomposition can be performed. We do not study the remedy for situations in which the NORTA step-1 correlation matrix is not feasible.

2.3 Retrospective Approximation for SRFPs

The SRFP is to solve a deterministic equation $g(r) = \rho$ for the unique root $r = \rho^*$, where $g(r)$ can only be estimated by a consistent (or an unbiased) estimate $\bar{y}(r)$ for any given value of r. Our RVG problem contains an SRFP, solving $n(n-1)/2$ one-dimensional equations $g_{ij}(\rho^*_{ij}) = \rho_{ij}$ for ρ^*_{ij}, $1 \le i < j \le n$. The root-finding function $g_{ij}(\cdot)$ defined in Equation (2) is a two-dimensional integral and hence may be difficult to compute numerically. However, $g_{ij}(\cdot)$ can be estimated easily by simulation, e.g., Equation (6) or (7). Chen and Schmeiser (1994a) provide other examples of the SRFP.

Chen and Schmeiser (1994b) propose retrospective approximation (RA) algorithms for SRFPs. RA iteratively solves a sequence of sample-path equations with increasing sample sizes and decreasing error tolerances. In each iteration, the sample-path equation is solved within the error tolerance; the root estimator is then a function of those solutions. Under proper conditions, the root estimator converges to the true root with probability one (w.p. 1).

To be more specific, we describe here the concept of sample-path equations. Let $\underline{\omega} = (\omega_1, ..., \omega_m)$ denote the random numbers used to obtain a consistent estimate $\bar{y}(r) = \bar{y}(r; \underline{\omega})$ of the function value $g(r)$ with sample size m; the components of $\underline{\omega}$ are generated independently. Holding $\underline{\omega}$ constant over all points r yields a sample-path approximation $\bar{y}(\cdot; \underline{\omega})$ to g. Given

$\underline{\omega}$, the sample-path approximation $\bar{y}(r;\underline{\omega})$ is a deterministic function of r. The sample-path equation

$$\bar{y}(r^*;\underline{\omega}) = \rho$$

then defines the random retrospective root r^*, which is an estimator of the true root ρ^*. Huber (1964) calls r^* an M-estimator. Here we define ω_l, $l = 1, ..., m$, as random numbers. This is a natural and safe choice in simulation. Depending on the model, however, ω_l can be chosen differently for greater computational efficiency. Since $\underline{\omega}$ is generated only once in solving the sample-path equation, any choice of ω_l must be functionally independent of r in $\bar{y}(r;\underline{\omega})$. Ideally, a large proportion of the computation of $\bar{y}$ should be contained in $\underline{\omega}$ so that these computations need be performed only once for many points of r. Alternative choices of ω_l for the RVG application are discussed in Section 3.

RA iteratively solves a sequence of sample-path equations

$$\bar{y}(r_k^*;\underline{\omega}_k) = \rho, \tag{4}$$

for $k = 1, 2, ...$, where the kth retrospective root r_k^* is the true root of Equation (4). The vector of random numbers $\underline{\omega}_k = (\omega_{k,1}, \omega_{k,2}, ..., \omega_{k,m_k})$ is selected independently for each k and the sample-size sequence $\{m_k\}$ is strictly increasing. At each iteration k, a solution r_k is found within error tolerance ϵ_k—i.e., $|r_k - r_k^*| < \epsilon_k$—where $\{\epsilon_k\}$ is a positive sequence converging to zero. We call r_k the kth retrospective solution. The root estimator at iteration k is then defined as $\bar{r}_k = \sum_{t=1}^k m_t r_t / \sum_{t=1}^k m_t$, a weighted average of the solutions r_1,..., r_k.

A natural stopping rule for RA centers on a standard-error estimate for the current root estimator. Assumptions that retrospective solutions r_k are unbiased independent estimators of the desired root ρ^* and that their variances are inversely proportional to the sample size m_k (i.e., $m_k \mathrm{Var}(r_k) = \nu^2$ for some constant $\nu > 0$) for each k yield two useful results: (i) After k retrospective iterations $\bar{r}_k$ is an unbiased estimator of ρ^*; (ii) Its variance is $\mathrm{Var}(\bar{r}_k) = \nu^2 / \sum_{t=1}^k m_t$ and hence has an unbiased estimator

$$\widehat{\mathrm{Var}}(\bar{r}_k) \;=\; \hat{\nu}^2 / \sum_{t=1}^k m_t \;=\; \left[\sum_{t=1}^k m_t r_t^2 - (\sum_{t=1}^k m_t)\bar{r}_k^2\right]\left[(k-1)\sum_{t=1}^k m_t\right]^{-1}. \tag{5}$$

To obtain such a natural stopping rule, after each RA iteration the standard-error estimate $\sqrt{\widehat{\mathrm{Var}}(\bar{r}_k)}$ is computed and the algorithm stopped when the estimate is within a user-specified precision σ_0. To avoid stopping prematurely because of unstable variance estimates in early iterations, at least four RA iterations are suggested.

The procedure for the family of generic RA algorithms is given below.

Family of RA Algorithms: solve the equation $g(r) = \rho$ for the root $r = \rho^*$.

Given: algorithm components:

1. an initial sample size m_1 and a rule for successively increasing m_k for $k \geq 2$,
2. a rule for computing an error-tolerance sequence $\{\epsilon_k\}$ that goes to zero as k goes to infinity, and
3. a numerical method for solving Equation (4) to within a specified error ϵ_k for each k.

Do:

step 0. Initialize the retrospective iteration number $k = 1$, the initial sample size m_1 and the initial error tolerance ϵ_1.

step 1. Independently generate $\underline{\omega}_k$.

step 2. Solve Equation (4) to obtain a solution r_k that satisfies $|r_k - r_k^*| < \epsilon_k$ using a deterministic root-finding method.

step 3. Compute $\bar{r}_k = \sum_{t=1}^{k} m_t r_t / \sum_{t=1}^{k} m_t$ and its variance estimate $\widehat{\mathrm{Var}}(\bar{r}_k)$.

step 4. If $\sqrt{\widehat{\mathrm{Var}}(\bar{r}_k)} < \sigma_0$, stop. Otherwise, compute m_{k+1} and ϵ_{k+1}, set $k \longleftarrow k+1$, and go to step 1.

Theorem 2 of Chen and Schmeiser (2001) shows that if $\sum_{k=1}^{\infty} \mathrm{E}|r_k^* - \rho^*| < \infty$, then the root estimator $\bar{r}_k$ converges to ρ^* w.p. 1 as k goes to infinity. Theorem 2 also holds even if the absolute difference is changed to the squared difference. The squared-difference condition seems likely to be satisfied in practice. For example, Lemma 3 of Chen and Schmeiser (2001) shows that the retrospective root r_k^* converges to ρ^* w.p. 1 if the root-finding function g is nondecreasing, the root ρ^* is unique, and the sample-path approximation $\bar{y}(r; \underline{\omega}_k)$ converges uniformly to $g(r)$ in r w.p. 1 as the sample size m_k goes to infinity (i.e., for every $\varepsilon > 0$, w.p. 1 there exists a positive integer $N(\varepsilon, \underline{\omega}_k)$ such that if $m_k > N(\varepsilon, \underline{\omega}_k)$ for some k, then $|\bar{y}(r; \underline{\omega}_k) - g(r)| < \varepsilon$ for all $r \in [-1, 1]$). Therefore, when k, and hence m_k, is large, $\mathrm{E}(r_k^* - \rho^*)^2$ decreases as m_k grows, e.g., $\mathrm{E}(r_k^* - \rho^*)^2 = \alpha / m_k$ for some constant $\alpha > 0$, as would be consistent with Lemma 2 of Chen and Schmeiser (2001). Then $\sum_{k=1}^{\infty} \mathrm{E}(r_k^* - \rho^*)^2$ is finite if the sample-size sequence is taken to be $m_{k+1} = c_1 m_k$, $c_1 > 1$, as recommended below.

A specific RA algorithm, called bounding RA, specifies the three algorithm components as follows: (1) Increase the sample size by a factor of $c_1 > 1$ at each iteration, i.e., $m_{k+1} = c_1 m_k$;

(2) Use a large finite $\{\epsilon_k\} : \epsilon_1 = 10^{50}$ and $\epsilon_{k+1} = \epsilon_k/\sqrt{c_1}$, so that essentially the numerical search stops when the first bounding interval is found; (3) At RA iteration k, solve the sample-path equation by the regula falsi search to find a retrospective solution r_k within the error tolerance ϵ_k. Such a method requires a bounding interval that brackets the retrospective root r_k^*. To find a bounding interval, an increment search starts at point $\bar{r}_{k-1}$ (where $\bar{r}_0 = r_0$, an initial root guess) and then moves in an increment or decrement of the initial step size δ_k. The process is repeated until the function changes sign, where the step size is doubled after each function evaluation for accelerating the search (see step 3.4 in the Appendix). At RA iteration k, the optimal initial step size δ_k would be $|\bar{r}_{k-1} - r_k^*|$ if r_k^* were known. Because $|\bar{r}_{k-1} - r_k^*|$ is proportional to $[\mathrm{Var}(\bar{r}_{k-1} - r_k^*)]^{1/2}$, we can choose $\delta_k = c_2[\widehat{\mathrm{Var}}(\bar{r}_{k-1} - r_k^*)]^{1/2}$, $c_2 > 0$, for $k \geq 3$. Based on the assumptions for variances (retrospective solutions r_k are independent and their variances are inversely proportional to the sample size) and the additional assumption $r_k^* = r_k$, $\mathrm{Var}(\bar{r}_{k-1} - r_k^*)$ equals $\nu^2[(\sum_{t=1}^{k-1} m_t)^{-1} + m_k^{-1}]$ and hence can be estimated based on Equation (5). Notice that the initial step size $\delta_k = c_2\hat{\nu}\sqrt{(\sum_{t=1}^{k-1} m_t)^{-1} + m_k^{-1}}$ needs to be computed before the kth RA iteration starts. Therefore, $\hat{\nu}$ has to be computed at the $(k-1)$st RA iteration, i.e., $\hat{\nu} = \sqrt{[\sum_{t=1}^{k-1} m_t r_t^2 - (\sum_{t=1}^{k-1} m_t)\bar{r}_{k-1}^2]/(k-2)}$. When $\hat{\nu}$ is inestimable for $k = 1, 2$, we can let $\delta_k = \delta_1 > 0$. The empirical results in Chen (1994) favor a small δ_1 (such as 0.0001). Larger values of c_2 accelerate the bound search but result in a bigger bounding interval. Setting $c_2 = 1$ seems reasonable. An upper limit may be set on the number of function evaluations in the numerical search to prevent a situation in which Equation (4) has no root or multiple roots during early retrospective iterations. We discuss below the choice of the sample-size and the error-tolerance sequences.

RA solves a sequence of sample-path equations with increasing sample sizes $\{m_k\}$. The reason for increasing m_k is as follows: In early RA iterations, many points are examined and hence m_k should be small. In later iterations, very few points are examined because the previous iterations provide a good initial guess $\bar{r}_{k-1}$ of r_k^* and hence m_k should grow large. Smaller values of m_1 and c_1 provide more times at which to stop the algorithm. To obtain only integer values of m_k, we typically use $c_1 = 2$. Any small value of m_1, including $m_1 = 1$, is acceptable; some numerical root-finding methods might be able to use standard-error information about $\bar{y}(\cdot; \underline{\omega}_1)$, in which case a larger value of m_1 might be useful. But m_1 should be small because at the first iteration nothing is known about the location of the retrospective root r_1^*, causing the root-finding method to examine many points r, with each examination requiring m_1 observations for $\bar{y}(r; \cdot)$. Figures 3 and 4 in Section 4 show that

RA is more efficient with small values of m_1, even when the initial point r_0 is very close to the root ρ^*.

The rule for determining the decreasing error-tolerance sequence $\{\epsilon_k\}$ can also take many forms. Since the sample sizes $\{m_k\}$ increase by a factor of c_1, then the assumption that $\mathrm{Var}(r_k)$ decreases with increasing sample size m_k suggests the form $\epsilon_k = \epsilon_{k-1}/\sqrt{c_1}$. The problem is then to specify an appropriate ϵ_1. If ϵ_1 is chosen too small, the root-finding method wastes computation finding a solution r_k close to r_k^*, which might not be close to ρ^* because of the small sample size. Because we are assuming that no prior information is available about sampling error, a value of ϵ_1 scaled to the problem at hand is not known. Bounding RA uses a very large value of $\epsilon_1 = 10^{50}$; essentially, the numerical search stops when the bounding interval is found. This eliminates the need to select a problem-dependent value. The error-tolerance logic is then reduced to being a device to prove convergence. Chen (1994) shows that bounding RA converges quickly in practice, regardless of the values of the algorithm parameters m_1, c_1, c_2, and δ_1.

3. Method

Here we propose a method for generating random vectors $\underline{X} = (X_1, ..., X_n)$ with marginal distributions $F_1, ..., F_n$ and correlation matrix $(\rho_{ij})_{i,j=1}^n$. Our generation method uses the NORTA approach, which transforms a standard-normal random vector $(Z_1, ..., Z_n)$ with correlation matrix $(\rho_{ij}^*)_{i,j=1}^n$ into the desired random vector $(X_1, ..., X_n)$, as discussed in Section 2.2. We use the bounding RA algorithm (Section 2.3) to find the values of ρ_{ij}^* so that the generated random vector $\underline{X}$ has the desired correlations $\rho_{ij}, 1 \le i, j \le n$. That is, bounding RA solves $n(n-1)/2$ one-dimensional equations $g_{ij}(\rho_{ij}^*) = \rho_{ij}$ for ρ_{ij}^*, $1 \le i < j \le n$, where g_{ij} is defined in Equation (2). Specifically, the framework of our generation method is:

NORTA RVG Methods: generate s observations of the random vector $\underline{X} = (X_1, ..., X_n)$ with marginal distributions $F_1, ..., F_n$ and correlation matrix $(\rho_{ij})_{i,j=1}^n$ as follows.

Given:

1. inputs specifying the random-vector characteristics described in the definition of the RVG problem (Section 1), and

2. values for m_1, $c_1 > 1$, $c_2 > 0$, $\delta_1 > 0$, and $\sigma_0 > 0$ in the bounding RA algorithm. (The default values are $m_1 = 40$, $c_1 = 2$, $c_2 = 1$, $\delta_1 = 0.0001$, and $\sigma_0 = 0.01$.)

Do:

(I) Use bounding RA simultaneously to solve $n(n-1)/2$ one-dimensional equations $g_{ij}(\rho^*_{ij}) = \rho_{ij}$ for ρ^*_{ij} until all root estimates $\hat{\rho}^*_{ij}$ have a standard-error estimate less than σ_0, using ρ_{ij} as the initial guess for $1 \leq i < j \leq n$. Let $\hat{\rho}_{ii} = 1$ and $\hat{\rho}^*_{ji} = \hat{\rho}^*_{ij}$ for $1 \leq i < j \leq n$.

(II) Apply NORTA to generate one observation of random vector $\underline{X} = (X_1, ..., X_n)$, using $(\hat{\rho}^*_{ij})_{i,j=1}^n$ as the correlation matrix for the standard-normal random vector; repeat this step s times to generate s observations.

We describe this RVG method in greater detail in the Appendix. Since step (II) is a direct application of NORTA, we discuss here only the implementation of step (I).

Four properties of g_{ij} are useful for root finding:

1. Function $g_{ij}(r)$ is nondecreasing for $-1 \leq r \leq 1$ and is continuous under mild conditions (see Theorems 1 and 2). Often, $g_{ij}(r)$ is a strictly increasing function even when the marginal distributions are discrete. Consider an example where $n = 2$ and both marginal distributions are Bernoulli with probability of success 0.5. Then, $g_{ij}(r) = 4[\mathrm{E}(X_1X_2) - (0.5)(0.5)] = -1 + 4\,P\{X_1X_2 = 1\} = -1 + 4\,P\{Z_1 \geq 0, Z_2 \geq 0\}$, where (Z_1, Z_2) are bivariate standard-normal with correlation r. Therefore, $g_{ij}(r)$ is a strictly increasing function of r. Other examples include the examples studied in Section 4.

2. The function $g_{ij}(r)$ lies on the origin for $r = 0$, below or on the 45-degree line for positive r and above or on the 45-degree line for negative r. This is because $|g_{ij}(r)| \leq |r|$ for any r in $[-1, 1]$, as proven in Lancaster (1957); equality holds only if F_i and F_j are both normal distributions (Mardia 1970, p. 33).

3. Function $g_{ij}(r)$ reaches its maximum $\bar{\rho}_{ij}$ when $r = 1$ and reaches its minimum $\underline{\rho}_{ij}$ when $r = -1$ (see Proposition 1). In general, if X_i and X_j have the same marginal distributions, the maximum positive target correlation that NORTA can achieve is $\bar{\rho}_{ij} = 1$. However, the minimum correlation $\underline{\rho}_{ij} = -1$ can be achieved (via NORTA) only when both F_i and F_j have a same *symmetric* shape such as uniform. If both F_i

and F_j are exponential, the value of $\underline{\rho}_{ij}$ is approximately -0.645. The proof can be found in Kendall and Stuart (1967, p. 568).

4. Function g_{ij} does not depend on the means and variances of the distributions F_i and F_j; hence, the root ρ^*_{ij} does not depend on the location and scale parameters of these distributions.

As mentioned in Section 2.2, Properties 1 and 3 imply that the solution ρ^*_{ij} of Equation (1) exists if the specified step-3 correlation ρ_{ij} is in the interval $[\underline{\rho}_{ij}, \bar{\rho}_{ij}]$. Even so, the step-1 correlation matrix $\Sigma_Z = (\rho^*_{ij})_{i,j=1}^n$ may not be nonnegative definite and hence may not be feasible. Ghosh and Henderson (2000) use the example of trivariate uniform random vectors to show that there are feasible $\Sigma_X = (\rho_{ij})_{i,j=1}^n$ for which the NORTA step-1 correlation matrix Σ_Z is not feasible. Cario and Nelson (1997) prove that if Σ_Z is nonnegative definite, then it follows from the NORTA transformation that Σ_X is also nonnegative definite. Putting these together, we have the following corollary.

Corollary 1. *If $g_{ij}(r)$, Σ_Z, and Σ_X satisfy the conditions*

1. *the function $g_{ij}(r)$ is nondecreasing and continuous for $-1 \le r \le 1$ for all $1 \le i < j \le n$,*

2. *each ρ_{ij} is within the interval $[\underline{\rho}_{ij}, \bar{\rho}_{ij}]$, $1 \le i < j \le n$, and*

3. *the solution correlation matrix $\Sigma_Z = (\rho^*_{ij})_{i,j=1}^n$ is positive definite, where each off-diagonal element ρ^*_{ij} satisfies Equation (1),*

then the random vector generated via a NORTA transformation is guaranteed to have marginal distributions $F_1, ..., F_n$ and a feasible correlation matrix Σ_X. Moreover, if function $g_{ij}(r)$ is strictly increasing, then Σ_Z is unique.

Condition 3, however, is difficult to verify. Usually we do not know whether Σ_Z is positive definite before solving for it. An alternative is to check whether the estimate $(\hat{\rho}^*_{ij})_{i,j=1}^n$ of Σ_Z is positive definite after the root finding in step (I). If it is not, the procedures proposed by Lurie and Goldberg (1998) and by Ghosh and Henderson (2000) can be applied to find a symmetric, positive definite approximation of Σ_Z for use in NORTA step 1 (Section 2.2).

Recall that solving equations $g_{ij}(\rho^*_{ij}) = \rho_{ij}$ for ρ^*_{ij}, $1 \le i < j \le n$, is an SRFP, in which equations are solved using estimates of function values. To implement bounding RA

for this SRFP, we need to define a consistent estimate of the function value $g_{ij}(r)$, where $r = \text{Corr}(Z_i, Z_j)$. For any r, $|r| \leq 1$, we can generate m pairs of bivariate standard-normal random variates (z_{il}, z_{jl}), $l = 1, ..., m$, each having correlation r. Then transform each pair (z_{il}, z_{jl}) into (x_{il}, x_{jl}), where $x_{il} = F_i^{-1}(\Phi(z_{il}))$, $l = 1, ..., m$, so that the marginal distributions of x_{il} and x_{jl} are F_i and F_j, respectively. A natural and consistent estimator of $g_{ij}(r) = \text{Corr}(X_{il}, X_{jl})$ is

$$\bar{y}_{ij}^C(r; \underline{\omega}) = \frac{\sum_{l=1}^m (x_{il} - \bar{x}_i)(x_{jl} - \bar{x}_j)}{\sqrt{[\sum_{l=1}^m (x_{il} - \bar{x}_i)^2]\,[\sum_{l=1}^m (x_{jl} - \bar{x}_j)^2]}} \quad , \tag{6}$$

where $\bar{x}_i = \sum_{l=1}^m x_{il}/m$. Here $\underline{\omega} = (\omega_1, ..., \omega_m)$ and the component ω_l, $l = 1, ..., m$, in $\underline{\omega}$ is the set of random numbers used to generate (z_{il}, z_{jl}). Other choices of ω_l are discussed later in this section. Another natural estimator $\sum_{l=1}^m (x_{il} - \mu_i)(x_{jl} - \mu_j)/(m\sigma_i\sigma_j)$, which uses the population means and variances (assumed easy to compute), is unbiased. However, the consistent estimator $\bar{y}_{ij}^C(r; \underline{\omega})$ is asymptotically more efficient than the unbiased estimator (Lehmann 1983, p. 441).

The control-variate variance-reduction technique (Law and Kelton 2000, p. 604) can be used to improve the crude estimate $\bar{y}_{ij}^C(r; \underline{\omega})$. Let $\hat{\theta}_{ij}(r; \underline{\omega})$ be the estimate of $\text{Corr}(Z_i, Z_j)$ based on observations (z_{il}, z_{jl}) for $l = 1, ..., m$, i.e.,

$$\hat{\theta}_{ij}(r; \underline{\omega}) = \frac{\sum_{l=1}^m (z_{il} - \bar{z}_i)(z_{jl} - \bar{z}_j)}{\sqrt{[\sum_{l=1}^m (z_{il} - \bar{z}_i)^2]\,[\sum_{l=1}^m (z_{jl} - \bar{z}_j)^2]}} \quad ,$$

where $\bar{z}_i = \sum_{l=1}^m z_{il}/m$. Then $\hat{\theta}_{ij}(r; \underline{\omega})$ can be an internal control variate for $\bar{y}_{ij}^C(r; \underline{\omega})$ because the expected value of $\hat{\theta}_{ij}(r; \underline{\omega})$ is asymptotically r, (z_{il}, z_{jl}) and (x_{il}, x_{jl}) are correlated, and the observations (z_{il}, z_{jl}), $l = 1, ..., m$, are already generated. The control-variate estimate of $g_{ij}(r)$ therefore has the form

$$\bar{y}_{ij}(r; \underline{\omega}) = \bar{y}_{ij}^C(r; \underline{\omega}) - \beta[\hat{\theta}_{ij}(r; \underline{\omega}) - r], \tag{7}$$

where $\beta = \widehat{\text{Cov}}[\,\bar{y}_{ij}^C(r; \underline{\omega}),\, \hat{\theta}_{ij}(r; \underline{\omega})\,]\ /\ \widehat{\text{Var}}[\,\hat{\theta}_{ij}(r; \underline{\omega})\,]$. The estimates $\widehat{\text{Cov}}[\,\bar{y}_{ij}^C(r; \underline{\omega}),\, \hat{\theta}_{ij}(r; \underline{\omega})\,]$ and $\widehat{\text{Var}}[\,\hat{\theta}_{ij}(r; \underline{\omega})\,]$ can be computed by micro/macro replications—a number of independent macro runs of multiple micro replications. At RA iteration k, the sample size is m_k. If 20 macro replications are used, then each macro run uses $\lfloor m_k/20 \rfloor$ micro replications to obtain one paired observation of $(\,\bar{y}_{ij}^C(r; \underline{\omega}),\, \hat{\theta}_{ij}(r; \underline{\omega})\,)$. After 20 runs, we can use the 20 independent paired observations of $(\,\bar{y}_{ij}^C(r; \underline{\omega}),\, \hat{\theta}_{ij}(r; \underline{\omega})\,)$ to estimate $\text{Var}[\,\hat{\theta}_{ij}(r; \underline{\omega})\,]$ and $\text{Cov}[\,\bar{y}_{ij}^C(r; \underline{\omega}),\, \hat{\theta}_{ij}(r; \underline{\omega})\,]$. If 20 macro replications are used at each RA iteration, an initial sample size of $m_1 \geq 40$ is recommended, resulting in at least two micro replications, for

generation of ($\bar{y}^C_{ij}(r;\underline{\omega})$, $\hat{\theta}_{ij}(r;\underline{\omega})$). (See also Section 4.) We do not set the control variate to be the unbiased estimate $\sum_{l=1}^{m} z_{il}z_{jl}/m$ of r because $\hat{\theta}_{ij}(r;\underline{\omega})$ can reduce both bias and variance of the function-value estimate.

Now we discuss solving the $n(n-1)/2$ equations $g_{ij}(\rho^*_{ij}) = \rho_{ij}$ for ρ^*_{ij} using the bounding RA algorithm. As mentioned in Section 2.2, each of these equations is independent of the others, is therefore one-dimensional, and hence can be solved separately. Solving all of these equations simultaneously, however, saves time on generating Monte Carlo observations. For example, to solve the $(n-1)$ equations $g_{1j}(\rho^*_{1j}) = \rho_{1j}$ for $j = 2, ..., n$, the consistent estimates $\bar{y}_{11}$, $\bar{y}_{12}$,...,$\bar{y}_{1n}$ need to be computed, each requiring m observations of X_1. If these equations are solved separately, the m observations of X_1 will be generated $(n-1)$ times for the $(n-1)$ equations. If we solve them simultaneously, we generate m observations of X_1 (as well as X_2,...,X_n) only once for use in all the equations. The drawback is that RA will not stop until all the $n(n-1)/2$ equations satisfy the standard-error tolerance σ_0, even if some of the equations have already reached the tolerance. Moreover, at RA iteration k, the root search (step 2 of RA in Section 2.3) will not stop until all root estimates are within the error tolerance ϵ_k. Another drawback is that in early RA iterations, the candidate standard-normal correlation matrix may not be Cholesky decomposable and hence RA may fail to generate standard-normal random vectors to estimate the function value $g_{ij}(\cdot)$. To fix this latter drawback, when a decomposition error occurs, we delete one column and one row from the candidate correlation matrix, starting from the right, until the reduced matrix is decomposable. Then we estimate the function values associated with the reduced matrix using Cholesky decomposition and estimate the other function values individually, instead of simultaneously. In later RA iterations the candidate matrix is close to $(\rho^*_{ij})_{i,j=1}^n$ and hence should be decomposable, unless $(\rho^*_{ij})_{i,j=1}^n$ is, or is nearly, not decomposable. In this case, $(\rho^*_{ij})_{i,j=1}^n$ should be adjusted to make it decomposable using the methods discussed in Lurie and Goldberg (1998) and Ghosh and Henderson (2000).

For this particular application, we slightly modify bounding RA for better efficiency. First, at RA iteration k, we require that the sample size m_k be big enough that for a given $\underline{\omega}_k$, the initial point $\bar{r}^{ij}_{k-1}$ satisfies $|\bar{y}_{ij}(\bar{r}^{ij}_{k-1};\underline{\omega}_k)| \leq |\bar{r}^{ij}_{k-1}|$. (Here $\bar{r}^{ij}_{k-1}$ is the RA estimate of ρ^*_{ij} after $k-1$ iterations.) Otherwise, m_k more observations are generated until the inequality holds. (See step 3.3 in the Appendix.) This requirement is based on the property that $|g_{ij}(r)| \leq |r|$ for any r in $[-1, 1]$. The precision of $\bar{y}_{ij}(\bar{r}^{ij}_{k-1};\underline{\omega}_k)$ is made to reflect this property. Therefore, from iteration to iteration, the sample size may increase by a factor

larger than c_1. Secondly, the sample-path approximation is defined as the crude estimate $\bar{y}_{ij}^C$ of Equation (6) if $m_k < 40$ and as the more efficient control-variate estimate $\bar{y}_{ij}$ of Equation (7) otherwise. (See step 3.2 in the Appendix.) As mentioned earlier, the control-variate coefficient β is computed by micro/macro replications. If 20 macro replications are used, the initial sample size m_k must be at least 40. Therefore, if $m_k < 40$, the crude estimate is used. Setting $m_1 = 40$ seems reasonable. On one hand, small m_1 seems to work better as shown in Section 4. On the other hand, if the initial sample size m_1 is so small that early RA iterations use crude estimates and later iterations use control-variate estimates, the variance of $\bar{r}_k^{ij}$ is overestimated and hence the bounding RA runs longer than it should. (For example, if $m_1 = 20$, the first retrospective solution r_1^{ij} has no control variate but the second retrospective solution r_2^{ij}, with $m_2 = 40$, has. At RA iteration 2, the root estimate is $\bar{r}_2^{ij} = (m_1 r_1^{ij} + m_2 r_2^{ij})/(m_1 + m_2)$. Based on the assumptions that both r_1^{ij} and r_2^{ij} are independent and unbiased and that their variances are inversely proportional to the sample size, $\mathrm{Var}(\bar{r}_2^{ij}) = (m_1\alpha_1 + m_2\alpha_2)/(m_1 + m_2)^2$, where $\mathrm{Var}(r_1^{ij}) = \alpha_1/m_1$, $\mathrm{Var}(r_2^{ij}) = \alpha_2/m_2$, and $\alpha_1 > \alpha_2 > 0$ because r_1^{ij} is computed without using control variates. Under the same assumptions, the estimate $\{\sum_{t=1}^2 m_t (r_t^{ij})^2 - (\sum_{t=1}^2 m_t)(\bar{r}_2^{ij})^2\}/\{(2-1)(m_1+m_2)\}$ of $\mathrm{Var}(\bar{r}_2^{ij})$ has mean $(m_2\alpha_1 + m_1\alpha_2)/(m_1+m_2)^2 > (m_1\alpha_1 + m_2\alpha_2)/(m_1+m_2)^2$ and hence, overestimates the true variance. The bias becomes negligible as the RA iteration number k increases.) To balance this trade-off, we recommend $m_1 = 40$.

In addition to m_1, the other bounding RA parameters need to be chosen. We set the initial solution $\bar{r}_0^{ij}$ (of ρ_{ij}^*) to ρ_{ij}, since ρ_{ij} is often a good guess of ρ_{ij}^* (see Section 4). The sample-size multiplier c_1, step-size multiplier c_2, and initial step size δ_1 are set arbitrarily to $c_1 = 2$, $c_2 = 1$, and $\delta_1 = 0.0001$ in our application. We do not study the effect of these parameter values here, but point out that in our previous work they have yielded good results (Chen 1994). The standard-error tolerance σ_0 is a user-dependent parameter. Smaller values of σ_0 provide more precise estimates of the standard-normal correlations ρ_{ij}^* and hence the generated NORTA random vectors have more accurate correlations. The tradeoff is that the root finding requires more computation time. One point concerning the root-finding error is that its magnitude satisfies $|g_{ij}(\rho_{ij}^* + \zeta) - \rho_{ij}| \leq |\zeta|$ for any ζ such that $|\rho_{ij}^* + \zeta| \leq 1$; hence, controlling the error in the function value does not guarantee meeting the error tolerance for the root.

To avoid an infinite search in solving the sample-path equation, a maximum number *maxit* of search points is set. At iteration k, Bounding RA solves Equation (4) until a

solution within error tolerance ϵ_k is obtained. For early iterations, Equation (4) may have zero or multiple roots. Hence, if the numerical method fails to meet the error-tolerance stopping criterion within *maxit* searches, the sample size m_k is increased by a factor of c_1 to define another sample-path equation with a larger sample size but with the same initial random-number seed. This new equation is then solved using the same initial point until the error tolerance is satisfied. (See steps 3.4 and 3.6 in the Appendix.) In this case, m_k is larger than $c_1 m_{k-1}$. For later iterations, the sample size m_k is large, making $\bar{y}_{ij}(\cdot;\underline{\omega}_k)$ close to the nondecreasing function $g_{ij}(\cdot)$ and hence, $\bar{y}_{ij}(\cdot;\underline{\omega}_k)$ crosses the level ρ_{ij} at a single point or a single interval. Solving Equation (4) to within ϵ_k accuracy is then easy.

Next, we discuss the choices of $\underline{\omega}$ and the convergence of RA in turn. Recall that $\underline{\omega} = (\omega_1, ..., \omega_m)$ must be functionally independent of all candidate step-1 correlations r^{ij}, $1 \leq i < j \leq n$, because $\underline{\omega}$ needs to be generated only once to solve the $n(n-1)/2$ sample-path equations (Section 2.3). There are three choices for each component ω_l of $\underline{\omega}$ for this RVG application. In order of increasing efficiency, they are (1) the random number seeds $(s_{1l}, ..., s_{nl})$, (2) the random numbers $(u_{1l}, ..., u_{nl})$ used to generate n independent standard-normal observations $(v_{1l}, ..., v_{nl})$, which are transformed into a set of correlated standard-normal observations $(z_{1l}, ..., z_{nl})$ with correlation matrix $(r^{ij})_{i,j=1}^n$ using the Cholesky decomposition, and (3) $(v_{1l}, ..., v_{nl})$. Because at each RA iteration, $\underline{\omega}$ is generated only once, higher levels of ω_l lead to less computation time. The values of $\underline{\omega}$ need to be stored for later evaluations of $\bar{y}$. In our RVG method, we choose the random-number seeds for ω_l. Despite its inefficiency, this choice has the advantage that it requires saving $\omega_1 = (s_{11}, ..., s_{n1})$ only, instead of the whole $\underline{\omega}$, because ω_2,..., ω_m are the succeeding seeds in the n random-number streams starting at $s_{11}, ..., s_{n1}$. For other choices of $\underline{\omega}$, the bookkeeping for all the components of $\underline{\omega}$ may become a substantial drain on the computer's storage capacity when the sample size m is large.

The retrospective concept in this RVG application is that at iteration k, only one sample $\underline{\omega}_k$ is used to solve the sample-path equation (4). That is, one sample of random-number seeds $\underline{\omega}_k$ is used to do the following: (i) generate m_k observations of the standard-normal random vector with candidate correlation matrix $(r^{ij})_{i,j=1}^n$ to compute $\bar{y}(r^{ij};\underline{\omega}_k)$; (ii) adjust the candidate matrix until all sample-path equations meet the error tolerance ϵ_k. Hence, at iteration k, the $\bar{y}$'s are evaluated at different points of interest but each with the same $\underline{\omega}_k$ (see steps 2 and 3 in the Appendix).

The convergence requirements for RA algorithms are discussed in Section 2.3. By Lemma 3 and Theorem 2 of Chen and Schmeiser (2001), the RA estimate $\bar{r}_k^{ij}$ converges to the true

root ρ_{ij}^* w.p. 1 if ρ_{ij}^* is unique and the function-value estimate $\bar{y}_{ij}(r;\cdot)$ converges uniformly to $g_{ij}(r)$ in r w.p. 1. For a simple example that satisfies the above conditions, we let $\bar{y}_{ij}(r;\underline{\omega})$ be the natural unbiased correlation estimate, i.e., $\sum_{l=1}^{m}(x_{il}-\mu_i)(x_{jl}-\mu_j)/(m\sigma_i\sigma_j)$. Assume that the distributions of X_i and X_j are same as the distribution of Z^3, where Z is a standard-normal random variable. Then we can show that the correlation of X_i and X_j is $(2r^3+3r)/5$, where X_i and X_j are defined via a NORTA transformation with bivariate normal correlation r. Hence, $g_{ij}(r)=(2r^3+3r)/5$, a strictly increasing function for $r\in[-1,1]$. Therefore, the root ρ_{ij}^* of equation $g_{ij}(\rho_{ij}^*)=\rho_{ij}$ is unique. We can also show that the unbiased correlation estimate $\bar{y}_{ij}(r;\cdot)$ defined above converges uniformly to $g_{ij}(r)$ w.p. 1. Hence, bounding RA converges for this example. Our RVG method uses the more efficient $\bar{y}_{ij}(r;\cdot)$ defined in Equation (7). In theory, $\bar{y}_{ij}(r;\cdot)$ may not converge uniformly for every marginal distributions of X_i and X_j. On the other hand, despite the lack of convergence proof, bounding RA performs well in our empirical results as shown in Section 4.

4. Empirical Results

Here we conduct two simulation experiments to evaluate the accuracy and efficiency of the RVG method proposed in Section 3. In the first experiment, we study the convergence speed of the RVG method and the variance-reduction effect of the internal control variate discussed in Section 3. Observations of the bivariate random vector (i.e., $n=2$) with uniform marginal distributions and correlation ρ_{12} are generated. For the special case of uniform random vectors, the step-1 correlation ρ_{ij}^* and the step-3 correlation ρ_{ij} satisfy the closed-form relationship: $\rho_{ij}^*=2\sin(\rho_{ij}\pi/6)$ for $1\le i,j\le n$ (Li and Hammond 1975). Because the value of ρ_{12}^* is known, we can estimate the mse of $\hat{\rho}_{12}^*$ as a function of the mean number N of observations generated to compute $\hat{\rho}_{12}^*$. The performance measure is the generalized mse, i.e., $(N)(\text{mse})$. As $N\to\infty$, the generalized mse approaches the asymptotic constant of the estimate $\hat{\rho}_{12}^*$. In the second experiment, we generate observations of random vectors with a variety of marginal distributions, estimate their correlations, and compare these estimates to the target values ρ_{ij}. Notice that in these two experiments, the distribution location and scale parameters are unimportant since the correlation measure is independent of them.

Our empirical results in both experiments show that bounding RA converges within reasonable computation time. The control-variate variance-reduction technique reduces the generalized mse significantly, ranging from a factor of 3.5 to a factor of 10. The control-

variate technique also increases the robustness of bounding RA since the convergence speed is good regardless of the choice of the initial sample size m_1. The bounding RA algorithm efficiency strongly depends on the type of marginal distributions specified. For example, as the noncentrality parameter of the t random vector increases, the difference between the root ρ_{ij}^* and initial guess ρ_{ij} increases. Hence, bounding RA requires more iterations (i.e., N is bigger) to satisfy the stopping rule. For the same reason, the smaller the shape parameter value of the gamma random vector, the more time-consuming the root finding. We describe the two experiments as follows.

In the first experiment, 20 simulation runs of 1000 replications each are made on a Sun SparcStation 2; at each replication, the equation $g_{12}(\rho_{12}^*) = \rho_{12}$ is solved to generate one bivariate uniform(0,1) random vector. The mse of $\hat{\rho}_{12}^*$ is estimated, as well as the standard error of the mse estimate. As an index of the computational effort, the mean number N of random vectors generated to compute $\hat{\rho}_{12}^*$ at each replication is also estimated. On our machine, $N = 10,000$ corresponds to about 0.2 seconds of CPU time. The parameter values of the bounding RA algorithm are chosen without tuning to be $c_1 = 2$, $c_2 = 1$, and $\delta_1 = 0.0001$.

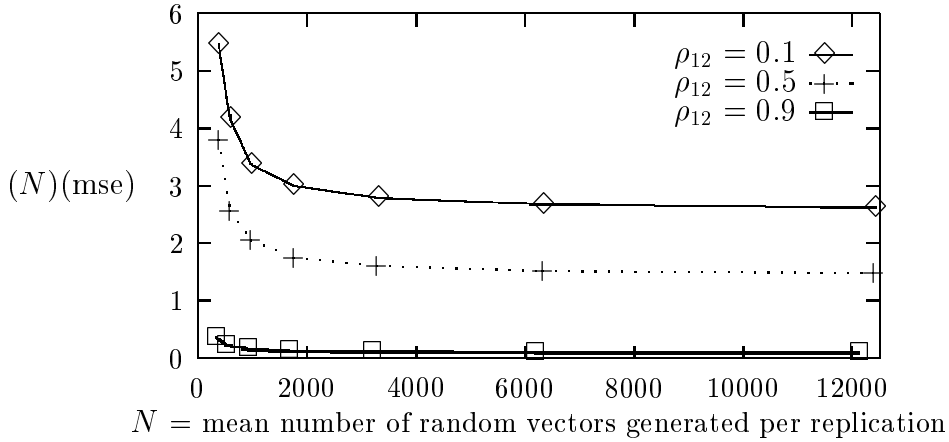

Figure 1: Without Control Variate: Three Generalized-mse Plots for the Bivariate Random Vector with Uniform Marginal Distributions, Correlations $\rho_{12} = 0.1$, 0.5, and 0.9, and $m_1 = 10$

Figures 1 and 2 plot the generalized mse as a function of N for the three target correlations $\rho_{12} = 0.1, 0.5, 0.9$ and $m_1 = 10$, without and with control variates, respectively. (Only positive correlations are considered because when ρ_{12} changes sign, only the sign of ρ_{12}^* changes, not its magnitude.) Both figures show that bounding RA converges quickly in all three cases, faster for larger ρ_{12}. This is because $\bar{y}_{12}(\rho_{12}^*; \cdot)$ in Equation (7) has an asymptotic

variance proportional to $(1-\rho_{12}^2)^2$, which decreases as ρ_{12} increases. (Notice that the crude estimate $\bar{y}_{12}^C(r;\cdot)$ in Equation 6 has an asymptotic variance $(1-[g(r)]^2)^2$. See Lehmann 1983, p. 441.) Comparing Figures 1 and 2, we see that the internal control variate reduces the generalized mse significantly, by a factor of about 10 for $\rho_{12} = 0.1$, 8 for $\rho_{12} = 0.5$, and 3.5 for $\rho_{12} = 0.9$.

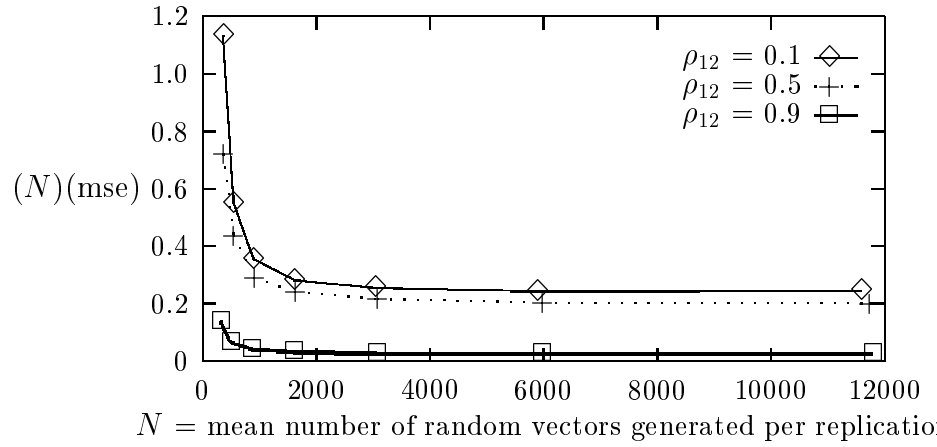

Figure 2: With Control Variate: Three Generalized-mse Plots for the Bivariate Random Vector with Uniform Marginal Distributions, Correlations $\rho_{12} = 0.1$, 0.5, and 0.9, and $m_1 = 10$

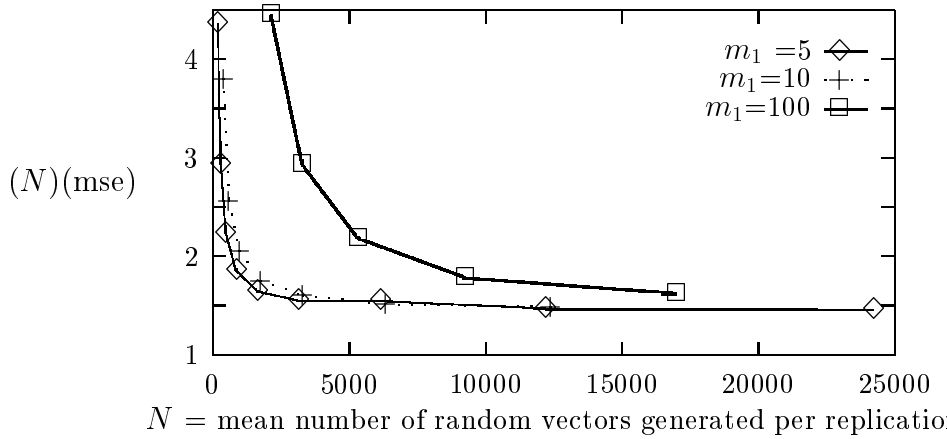

Figure 3: Without Control Variate: Three Generalized-mse Plots for the Bivariate Random Vector with Uniform Marginal Distributions, Correlation $\rho_{12} = 0.5$, and $m_1 = 5$, 10, and 100

Figures 3 and 4 illustrate the effect of the initial sample size m_1, without and with control variates, respectively. The generalized-mse curves for $\rho_{12} = 0.5$ and three different initial sample sizes $m_1 = 5$, 10, and 100 are plotted. Both figures show that small m_1 works better, even when the initial guess ρ_{12} (=0.5) is very close to the true root ρ_{12}^* (=0.5176). As shown in Figure 4, the control variate increases the robustness of bounding RA: the generalized-mse

is less sensitive to the choice of m_1, since the three generalized-mse curves for $m_1 = 5$, 10, and 100 are closer. In both figures, the generalized-mse curve for $m_1 = 100$ is much higher than for $m_1 = 5$ and 10; hence, a smaller value of m_1 is recommended.

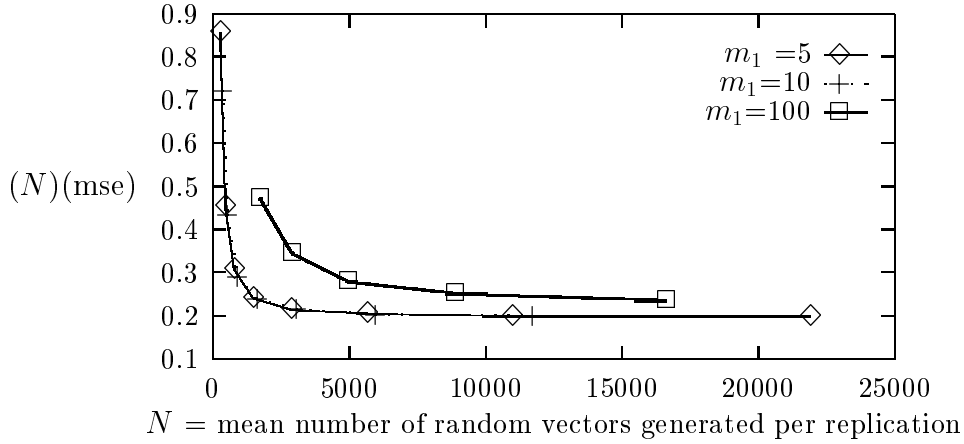

Figure 4: With Control Variate: Three Generalized-mse Plots for the Bivariate Random Vector with Uniform Marginal Distributions, Correlation $\rho_{12} = 0.5$, and $m_1 = 5$, 10, and 100

The second experiment is run for seven different combinations of marginal distributions F_1,..., F_n and dimension n, which fall into three cases:

Case 1: Bivariate random vector (X_1, X_2), where X_i has an exponential marginal distribution, $i = 1, 2$.

Case 2: Trivariate random vector (X_1, X_2, X_3), where the marginal distributions of X_i are the same for $i = 1, 2, 3$. We investigate five marginal distributions: (1) the exponential distribution, (2) the gamma distribution with shape parameter 5, (3) the gamma distribution with shape parameter 0.1, (4) the central t distribution with 3 degrees of freedom, and (5) the noncentral t distribution with 3 degrees of freedom and noncentrality parameter 10.

Case 3: Trivariate random vector (X_1, X_2, X_3), where each X_i has a different marginal distribution: X_1 is exponential, X_2 is gamma with shape parameter 7, and X_3 is triangular with minimum 0, mode 0, and maximum 100.

The target correlations $\rho_{ij}, 1 \leq i, j \leq n$, of the random vector $\underline{X} = (X_1, ..., X_n)$ are chosen from the set $\{0.1, 0.5, 0.9\}$. Of the $3^{n(n-1)/2}$ combinations, only feasible correlation matrices (at least positive definite) are studied to prevent computational errors during Cholesky

decomposition. Case 1 also considers negative correlations. The efficient ARTAFACTS software (Cario and Nelson 1998) can also handle Case 1 as well as Case 2 with autocorrelated correlation structure, i.e., $\rho_{12} = \rho_{23}$.

For each design point (specification of the marginal distributions and correlation matrix), 20 simulations are run on a Sun SparcStation 2. Each run solves for the step-1 correlations ρ_{ij}^* only once and uses them to generate 10^6 observations of the random vector $\underline{X}$. The estimates $\hat{\rho}_{ij}^*$ (average of 20 runs) and estimates $\hat{g}_{ij}$ of $\mathrm{E}[g_{ij}(\hat{\rho}_{ij}^*)]$ are computed (based on 20×10^6 observations). To evaluate the computational effort, the mean number N of random vectors generated for estimating ρ_{ij}^* and the CPU time (for estimating ρ_{ij}^* and generating 10^6 observations) for one simulation run are estimated. The bounding RA algorithm parameter m_1 is tuned to 40 since small m_1 seems better (see Figure 4) and 40 is the smallest possible m_1 when the control-variate estimate of $g_{ij}(\cdot)$ is used at every RA iteration (see Section 3). The standard-error tolerance σ_0 is set to 0.01. The other bounding RA parameters are assigned the same values as in the first experiment. To prevent RA from stopping prematurely, at least four RA iterations are required. Observations from each marginal distribution are generated as follows: We numerically compute the inverse of the noncentral t distribution function, where the distribution and density functions are evaluated using the code from Lenth (1989). The gamma-inverse and normal-inverse subroutines are from Best and Roberts (1975) and Beasley and Springer (1977), respectively. The exponential and triangular distribution functions have closed-form inverses. Simulation results are shown in Tables 1 through 7. For comparison, common random numbers are used for all design points. Only significant digits are listed.

Table 1: Simulation Results for Bivariate $\underline{X}$ with Exponential Marginal Distributions

ρ_{12}	$\hat{g}_{12}$	$\hat{\rho}_{12}^*$	N (1000's)	CPU (sec./run)
-0.5	-0.499	-0.722	15	19
-0.1	-0.099	-0.125	9	19
0.1	0.100	0.119	16	19
0.5	0.504	0.550	11	19
0.9	0.896	0.911	2	19

Table 1 shows the simulation results for Case 1. One million bivariate exponential random vectors are generated (with one-time root finding) within a few seconds. The estimated correlations $\hat{g}_{12}$ (with standard error around 0.002) of (X_1, X_2) are close to the target values ρ_{12}. As in Figure 2, the root finding with $\rho_{12} = 0.9$ has the lowest N and hence is most

efficient. Table 1 also shows the relation between ρ^*_{ij} (root) and ρ_{ij} (desired function value); the root estimates $\hat{\rho}^*_{12}$ are near ρ_{12}, except for $\rho_{12} = -0.5$. (Recall that the minimum correlation of a bivariate exponential random vector, defined via NORTA, is around -0.645.)

Tables 2 through 6 show the simulation results for Case 2 and Table 7 shows the results for Case 3. For Case 2, we see that the correlation estimates $\hat{g}_{ij}$ of the generated random vector are close to the target values ρ_{ij}. (The standard-error estimates for $\hat{g}_{ij}$ range from 0.001 to 0.005 for Tables 4 and 6 and from 0.0005 to 0.004 for the other tables. No clear bias was found after checking.) The root-finding computation time (evaluated based on N), in increasing order, is gamma with shape 5, exponential (equivalently, gamma with shape 1), central t with 3 degrees of freedom, gamma with shape 0.1, and noncentral t with noncentrality parameter 10 and 3 degrees of freedom. As we traverse this list, ρ^*_{ij} deviates increasingly from ρ_{ij} and therefore N increases. The RVG efficiency also depends on the computation required to invert the marginal distributions; recall that during the root search for ρ^*_{ij}, m_k random vectors must be generated at RA iteration k, while generation of each random vector requires inverting n marginal distributions. Table 7 for Case 3 shows that our RVG method also works well for random vectors with different marginal distributions.

Table 2: Simulation Results for Trivariate $\underline{X}$ with Exponential Marginal Distributions

$(\rho_{12}, \rho_{13}, \rho_{23})$	$(\hat{g}_{12}, \hat{g}_{13}, \hat{g}_{23})$	$(\hat{\rho}^*_{12}, \hat{\rho}^*_{13}, \hat{\rho}^*_{23})$	N (1000's)	CPU (sec./run)
(0.1, 0.1, 0.1)	(0.103, 0.099, 0.098)	(0.122, 0.118, 0.117)	32	26
(0.1, 0.1, 0.5)	(0.099, 0.100, 0.498)	(0.118, 0.119, 0.544)	31	26
(0.1, 0.1, 0.9)	(0.104, 0.105, 0.899)	(0.123, 0.124, 0.914)	23	26
(0.1, 0.5, 0.1)	(0.099, 0.499, 0.100)	(0.118, 0.545, 0.119)	31	26
(0.1, 0.5, 0.5)	(0.103, 0.503, 0.503)	(0.123, 0.549, 0.549)	23	26
(0.1, 0.9, 0.1)	(0.104, 0.901, 0.104)	(0.124, 0.916, 0.123)	19	25
(0.5, 0.1, 0.1)	(0.502, 0.101, 0.100)	(0.548, 0.120, 0.119)	22	26
(0.5, 0.1, 0.5)	(0.499, 0.103, 0.503)	(0.546, 0.122, 0.549)	30	26
(0.5, 0.5, 0.1)	(0.500, 0.501, 0.104)	(0.547, 0.548, 0.123)	28	26
(0.5, 0.5, 0.5)	(0.502, 0.501, 0.501)	(0.548, 0.547, 0.547)	29	26
(0.5, 0.5, 0.9)	(0.501, 0.499, 0.900)	(0.547, 0.545, 0.915)	19	25
(0.5, 0.9, 0.5)	(0.501, 0.899, 0.500)	(0.547, 0.914, 0.546)	20	26
(0.9, 0.1, 0.1)	(0.899, 0.099, 0.101)	(0.914, 0.119, 0.121)	18	26
(0.9, 0.5, 0.5)	(0.900, 0.501, 0.501)	(0.915, 0.547, 0.548)	21	25
(0.9, 0.9, 0.9)	(0.901, 0.900, 0.899)	(0.915, 0.915, 0.914)	6.1	26

Table 3: Simulation Results for Trivariate $\underline{X}$ with Gamma Marginal Distributions of Shape Parameter 5

$(\rho_{12},\ \rho_{13},\ \rho_{23})$	$(\hat{g}_{12}, \hat{g}_{13}, \hat{g}_{23})$	$(\hat{\rho}^*_{12}, \hat{\rho}^*_{13}, \hat{\rho}^*_{23})$	N (1000's)	CPU (sec./run)
(0.1, 0.1, 0.1)	(0.101, 0.101, 0.102)	(0.105, 0.105, 0.106)	7	99
(0.1, 0.1, 0.5)	(0.100, 0.099, 0.498)	(0.104, 0.102, 0.509)	6	99
(0.1, 0.1, 0.9)	(0.107, 0.105, 0.900)	(0.111, 0.109, 0.904)	4	98
(0.1, 0.5, 0.1)	(0.103, 0.503, 0.105)	(0.107, 0.514, 0.109)	6	98
(0.1, 0.5, 0.5)	(0.102, 0.500, 0.500)	(0.106, 0.510, 0.511)	5	98
(0.1, 0.9, 0.1)	(0.105, 0.900, 0.103)	(0.110, 0.903, 0.106)	4	98
(0.5, 0.1, 0.1)	(0.499, 0.103, 0.103)	(0.509, 0.107, 0.107)	6	99
(0.5, 0.1, 0.5)	(0.502, 0.107, 0.504)	(0.513, 0.111, 0.514)	5	98
(0.5, 0.5, 0.1)	(0.499, 0.504, 0.103)	(0.510, 0.514, 0.107)	5	98
(0.5, 0.5, 0.5)	(0.502, 0.501, 0.504)	(0.513, 0.511, 0.514)	5	98
(0.5, 0.5, 0.9)	(0.506, 0.506, 0.899)	(0.517, 0.516, 0.903)	4	98
(0.5, 0.9, 0.5)	(0.505, 0.902, 0.503)	(0.515, 0.905, 0.513)	4	98
(0.9, 0.1, 0.1)	(0.902, 0.103, 0.103)	(0.905, 0.107, 0.107)	5	99
(0.9, 0.5, 0.5)	(0.900, 0.500, 0.501)	(0.904, 0.510, 0.511)	5	98
(0.9, 0.9, 0.9)	(0.899, 0.900, 0.900)	(0.903, 0.903, 0.904)	2.4	99

Table 4: Simulation Results for Trivariate $\underline{X}$ with Gamma Marginal Distributions of Shape Parameter 0.1

$(\rho_{12},\ \rho_{13},\ \rho_{23})$	$(\hat{g}_{12}, \hat{g}_{13}, \hat{g}_{23})$	$(\hat{\rho}^*_{12}, \hat{\rho}^*_{13}, \hat{\rho}^*_{23})$	N (1000's)	CPU (sec./run)
(0.1, 0.1, 0.1)	(0.100, 0.098, 0.099)	(0.222, 0.216, 0.220)	4.7×100	188
(0.1, 0.1, 0.5)	(0.098, 0.098, 0.497)	(0.218, 0.218, 0.676)	3.1×100	72
(0.1, 0.1, 0.9)	(0.101, 0.101, 0.899)	(0.223, 0.222, 0.946)	3.1×100	81
(0.1, 0.5, 0.1)	(0.102, 0.502, 0.103)	(0.225, 0.680, 0.225)	3.4×100	107
(0.1, 0.5, 0.5)	(0.100, 0.495, 0.502)	(0.221, 0.676, 0.680)	2.6×100	30
(0.1, 0.9, 0.1)	(0.101, 0.900, 0.101)	(0.222, 0.947, 0.222)	2.4×100	9
(0.1, 0.9, 0.5)	(0.498, 0.102, 0.100)	(0.677, 0.224, 0.222)	2.9×100	63
(0.5, 0.1, 0.5)	(0.499, 0.100, 0.502)	(0.678, 0.220, 0.681)	3.5×100	115
(0.5, 0.5, 0.1)	(0.501, 0.502, 0.101)	(0.679, 0.682, 0.221)	2.8×100	44
(0.5, 0.5, 0.5)	(0.502, 0.497, 0.502)	(0.680, 0.677, 0.681)	1.7×100	145
(0.5, 0.5, 0.9)	(0.497, 0.497, 0.900)	(0.677, 0.677, 0.946)	1.3×100	107
(0.5, 0.9, 0.5)	(0.499, 0.897, 0.503)	(0.678, 0.945, 0.681)	1.2×100	103
(0.9, 0.1, 0.1)	(0.899, 0.101, 0.100)	(0.946, 0.222, 0.221)	2.8×100	55
(0.9, 0.5, 0.5)	(0.898, 0.497, 0.500)	(0.946, 0.677, 0.680)	1.2×100	95
(0.9, 0.9, 0.9)	(0.896, 0.894, 0.898)	(0.944, 0.943, 0.946)	8	193

Table 5: Simulation Results for Trivariate $\underline{X}$ with Central t Marginal Distributions of 3 Degrees of Freedom

$(\rho_{12}, \rho_{13}, \rho_{23})$	$(\hat{g}_{12}, \hat{g}_{13}, \hat{g}_{23})$	$(\hat{\rho}^*_{12}, \hat{\rho}^*_{13}, \hat{\rho}^*_{23})$	N (1000's)	CPU (sec./run)
(0.1, 0.1, 0.1)	(0.101, 0.100, 0.100)	(0.118, 0.119, 0.117)	9×10	41
(0.1, 0.1, 0.5)	(0.102, 0.101, 0.497)	(0.120, 0.119, 0.553)	6×10	14
(0.1, 0.1, 0.9)	(0.100, 0.098, 0.898)	(0.117, 0.116, 0.920)	34	208
(0.1, 0.5, 0.1)	(0.100, 0.492, 0.101)	(0.117, 0.557, 0.119)	1.6×100	101
(0.1, 0.5, 0.5)	(0.100, 0.494, 0.493)	(0.117, 0.557, 0.553)	1.0×100	51
(0.1, 0.9, 0.1)	(0.096, 0.894, 0.099)	(0.113, 0.920, 0.116)	24	203
(0.5, 0.1, 0.1)	(0.497, 0.099, 0.099)	(0.554, 0.118, 0.117)	1.1×100	56
(0.5, 0.1, 0.5)	(0.497, 0.099, 0.490)	(0.554, 0.117, 0.552)	1.5×100	86
(0.5, 0.5, 0.1)	(0.499, 0.493, 0.099)	(0.556, 0.560, 0.117)	1.3×100	73
(0.5, 0.5, 0.5)	(0.496, 0.492, 0.491)	(0.553, 0.556, 0.555)	1.4×100	81
(0.5, 0.5, 0.9)	(0.491, 0.486, 0.893)	(0.548, 0.547, 0.919)	3.2×100	206
(0.5, 0.9, 0.5)	(0.492, 0.895, 0.492)	(0.549, 0.921, 0.552)	4×10	2
(0.9, 0.1, 0.1)	(0.897, 0.098, 0.098)	(0.919, 0.117, 0.118)	4×10	2
(0.9, 0.5, 0.5)	(0.898, 0.491, 0.488)	(0.920, 0.555, 0.557)	7×10	28
(0.9, 0.9, 0.9)	(0.896, 0.892, 0.891)	(0.918, 0.917, 0.918)	13	194

Table 6: Simulation Results for Trivariate $\underline{X}$ where the Marginal Distributions are Noncentral t with Noncentrality Parameter 10 and 3 Degrees of Freedom

$(\rho_{12}, \rho_{13}, \rho_{23})$	$(\hat{g}_{12}, \hat{g}_{13}, \hat{g}_{23})$	$(\hat{\rho}^*_{12}, \hat{\rho}^*_{13}, \hat{\rho}^*_{23})$	N (1000's)	CPU (sec./run)
(0.1, 0.1, 0.1)	(0.101, 0.099, 0.097)	(0.161, 0.162, 0.157)	4×100	197
(0.1, 0.1, 0.5)	(0.099, 0.097, 0.500)	(0.158, 0.159, 0.623)	6×100	37
(0.1, 0.1, 0.9)	(0.099, 0.096, 0.896)	(0.159, 0.159, 0.935)	1.7×100	47
(0.1, 0.5, 0.1)	(0.100, 0.488, 0.099)	(0.159, 0.628, 0.160)	4×100	97
(0.1, 0.5, 0.5)	(0.100, 0.496, 0.497)	(0.159, 0.626, 0.629)	6×100	94
(0.1, 0.9, 0.1)	(0.097, 0.892, 0.097)	(0.156, 0.936, 0.156)	1.5×100	158
(0.5, 0.1, 0.1)	(0.506, 0.098, 0.098)	(0.624, 0.161, 0.161)	8×100	43
(0.5, 0.5, 0.1)	(0.501, 0.498, 0.101)	(0.618, 0.641, 0.163)	1×1000	135
(0.5, 0.5, 0.5)	(0.504, 0.492, 0.491)	(0.623, 0.628, 0.631)	1.0×1000	188
(0.5, 0.5, 0.9)	(0.495, 0.489, 0.894)	(0.616, 0.618, 0.937)	4×100	12
(0.5, 0.9, 0.5)	(0.511, 0.899, 0.510)	(0.629, 0.940, 0.635)	5×100	17
(0.9, 0.1, 0.1)	(0.900, 0.097, 0.096)	(0.937, 0.161, 0.163)	5×100	33
(0.9, 0.5, 0.5)	(0.899, 0.485, 0.480)	(0.936, 0.624, 0.630)	4×100	186
(0.9, 0.9, 0.9)	(0.890, 0.889, 0.893)	(0.930, 0.932, 0.935)	6×10	103

Table 7: Simulation Results for Trivariate $\underline{X}$ where X_1 is Exponential, X_2 is Gamma with Shape Parameter 7, and X_3 is Triangular with Minimum 0, Mode 0, and Maximum 100

$(\rho_{12},\, \rho_{13},\, \rho_{23})$	$(\hat{g}_{12}, \hat{g}_{13}, \hat{g}_{23})$	$(\hat{\rho}^*_{12}, \hat{\rho}^*_{13}, \hat{\rho}^*_{23})$	N (1000's)	CPU (sec./run)
(0.1, 0.1, 0.1)	(0.102, 0.103, 0.101)	(0.113, 0.115, 0.104)	5×10	57
(0.1, 0.1, 0.5)	(0.102, 0.103, 0.499)	(0.113, 0.116, 0.513)	24	56
(0.1, 0.1, 0.9)	(0.103, 0.103, 0.900)	(0.114, 0.116, 0.914)	12	55
(0.1, 0.5, 0.1)	(0.101, 0.500, 0.100)	(0.112, 0.546, 0.104)	8×10	58
(0.1, 0.5, 0.5)	(0.100, 0.500, 0.501)	(0.112, 0.546, 0.514)	24	55
(0.1, 0.9, 0.1)	(0.102, 0.897, 0.101)	(0.114, 0.954, 0.105)	5×10	116
(0.5, 0.1, 0.1)	(0.500, 0.103, 0.100)	(0.538, 0.116, 0.104)	4×10	56
(0.5, 0.1, 0.5)	(0.501, 0.102, 0.500)	(0.539, 0.115, 0.514)	3×10	56
(0.5, 0.5, 0.1)	(0.499, 0.498, 0.101)	(0.537, 0.543, 0.104)	6×10	57
(0.5, 0.5, 0.5)	(0.502, 0.499, 0.501)	(0.540, 0.545, 0.515)	21	55
(0.5, 0.5, 0.9)	(0.501, 0.501, 0.901)	(0.539, 0.546, 0.914)	12	56
(0.5, 0.9, 0.5)	(0.503, 0.899, 0.500)	(0.541, 0.956, 0.514)	12	104
(0.9, 0.1, 0.1)	(0.899, 0.100, 0.101)	(0.938, 0.113, 0.105)	19	56
(0.9, 0.5, 0.5)	(0.898, 0.502, 0.502)	(0.937, 0.548, 0.515)	22	56
(0.9, 0.9, 0.9)	(0.899, 0.898, 0.900)	(0.937, 0.955, 0.913)	4.5	105

5. Conclusions

We have proposed an algorithm to generate observations of the random vector—defined by the NORTA transformation—with specified marginal distributions and correlation matrix. Two major steps of this algorithm are as follows: (I) NORTA initialization: solve $n(n-1)/2$ equations simultaneously to compute the NORTA step-1 correlations using bounding RA, a generic SRFP algorithm that solves a sequence of sample-path equations with increasing sample sizes and decreasing error tolerances; (II) Generation via NORTA: generate observations using the NORTA approach where the step-1 correlations are computed from (I). We define an internal control variate to reduce the variance of function-value estimates in (I). Empirical results show that the control variate significantly improves the root-finding efficiency. Empirical results for a variety of marginal distributions show that the generated random vector has approximately the right correlations.

An alternative to bounding RA is to solve the equations numerically using off-the-shelf numerical root-finding algorithms, where function values are computed by numerical-integration methods. All equations can be solved separately for simplicity or simultaneously for efficiency. A comparison of the numerical and simulation approaches can be made based on the algorithm efficiency and accuracy. Consider the efficiency first. If the numerical and

simulation approaches solve all $n(n-1)/2$ equations simultaneously, then both approaches have the same order of computation; those computational tasks include two parts: one with order $O(n)$ and the other with order $O(n^2)$. The $O(n)$ part usually dominates unless n is large. The shortcoming of the simulation approach is that not all $n(n-1)/2$ equations can be solved simultaneously when the Cholesky decomposition fails in early RA iterations. In the worst case, all equations have to be solved separately and therefore the $O(n)$ component vanishes. In later RA iterations, the candidate standard-normal correlation matrix is close to the true standard-normal correlation matrix $(\rho^*_{ij})_{i,j=1}^n$ and hence should be decomposable unless $(\rho^*_{ij})_{i,j=1}^n$ is, or is nearly, not decomposable. The values of the NORTA step-1 correlations ρ^*_{ij} also affect the efficiencies of both approaches, but in a different way. In the simulation approach, larger values of $|\rho^*_{ij}|$ result in greater efficiency because the estimate of the root-finding function value then has a lower variance. In the numerical approach, smaller values of $|\rho^*_{ij}|$ are better. When the value of ρ^*_{ij} is close to 1 (or -1), the root-finding function is not smooth because the bivariate standard-normal density (in Equation 1) is almost zero everywhere outside the neighborhood of the 45-degree (or 135-degree) line. Therefore, the time required to compute the double-integral function value increases and the efficiency of the numerical approach decreases.

The accuracy of the simulation approach is less dependent on the value of ρ^*_{ij} than the numerical approach. When the magnitude of ρ^*_{ij} is close to 1, the numerical approach may be inaccurate if the more efficient Gaussian quadrature method is used to evaluate the double-integral function values. The numerical error may be so large that the root finding fails; for example, the function value evaluated by a Gaussian quadrature method may be higher than 1, but the function value is a correlation and hence should lie between -1 and 1. If the magnitude of ρ_{ij} is small, numerical approaches can be very efficient and accurate.

Appendix: NORTA RVG Methods

Purpose: generate s observations of the random vector $\underline{X} = (X_1, ..., X_n)$ with marginal distributions $F_1, ..., F_n$ and correlation matrix $(\rho_{ij})_{i,j=1}^n$.

Given:

1. Inputs specifying the random-vector characteristics:

(a) the dimension n of the random vector $\underline{X}$,

(b) a computer procedure to compute the inverse cumulative distribution functions F_1^{-1},...,F_n^{-1}, and

(c) a feasible correlation matrix $(\rho_{ij})_{i,j=1}^n$.

2. Values for the bounding RA algorithm parameters: initial sample size m_1, sample-size multiplier $c_1 > 1$, step-size multiplier $c_2 > 0$, initial step size $\delta_1 > 0$, and standard-error tolerance σ_0. (The default values are $m_1 = 40$, $c_1 = 2$, $c_2 = 1$, $\delta_1 = 0.0001$, and $\sigma_0 = 0.01$.)

Do:

(I) Use bounding RA simultaneously to solve the $n(n-1)/2$ one-dimensional equations $g_{ij}(\rho_{ij}^*) = \rho_{ij}$ for ρ_{ij}^*, $1 \le i < j \le n$, and obtain estimates $\hat{\rho}_{ij}^*$ of ρ_{ij}^* as follows.

0. Initialize $k = 0$. Set the initial error tolerance $\epsilon_1 = 10^{50}$, maximum number of numerical searches $maxit = 1000$, initial solutions $\bar{r}_0^{ij} = \rho_{ij}$, and initial step sizes $\delta_1^{ij} = \delta_1$ for all $1 \le i < j \le n$.
1. Let $k \longleftarrow k + 1$.
2. Randomly select n initial random-number seeds $(s_{k,1}, ..., s_{k,n})$ that define $\underline{\omega}_k = \{\omega_{k,1}, ..., \omega_{k,m_k}\}$, where $\omega_{k,1} = (s_{k,1}, ..., s_{k,n})$. For $i = 1, ..., m_k - 1$, the lth component in $\omega_{k,i+1}$ is the succeeding seed of the lth component in $\omega_{k,i}$ in the random-number stream starting with seed $s_{k,l}$, $l = 1, ..., n$.
3. Numerically solve the sample-path equations $\bar{y}_{ij}(r_k^{*ij}; \underline{\omega}_k) = \rho_{ij}$ for r_k^{*ij}, $1 \le i < j \le n$, as follows:
   3.1 Let $r^{ij} = \bar{r}_{k-1}^{ij}$ for $1 \le i < j \le n$ and $msave = m_k$. (Note: $\bar{r}_0^{ij} = \rho_{ij}$, initialized in step 0.)
   3.2 Compute $\bar{y}_{ij}(r^{ij}; \underline{\omega}_k)$ for $1 \le i < j \le n$, using Equation (7) for $m_k \ge 40$ or Equation (6) for $m_k < 40$. (Notice that 20 independent macro replications of $\lfloor m_k/20 \rfloor$ independent micro replications are used to compute the coefficient β in Equation 7). This is done as follows:
      (i) Perform Cholesky decomposition on the step-1 correlation matrix $(r^{ij})_{i,j=1}^n$ (note: $r^{ji} = r^{ij}$, $r^{ii} = 1$ for $1 \le i < j \le n$).
      (ii) If the decomposition succeeds, generate m_k observations of the random vector $(X_1, ..., X_n)$ via NORTA using the initial random-number seeds

$(s_{k,1}, ..., s_{k,n})$; then compute all $\bar{y}_{ij}(r^{ij}; \underline{\omega}_k)$ using Equation (7) or (6). If the decomposition fails, delete one column and one row, where the column and row indices are equal, from $(r^{ij})_{i,j=1}^{n}$, starting from the right, until the reduced matrix is decomposable. Suppose that b columns and b rows are deleted. Then implement NORTA to generate m_k observations of $(X_1, ..., X_{n-b})$ using the initial random-number seeds $(s_{k,1}, ..., s_{k,n-b})$ to compute $\bar{y}_{ij}(r^{ij}; \underline{\omega}_k)$ for $1 \le i < j \le n-b$. For the rest of the $\bar{y}_{ij}$'s, compute each $\bar{y}_{ij}$ separately by generating m_k paired observations of (X_i, X_j) with initial seeds $s_{k,i}$ and $s_{k,j}$.

3.3 The magnitude of $\bar{y}_{ij}(r^{ij}; \underline{\omega}_k)$ must be no larger than $|r|$. Therefore, if there is any pair (i, j) such that $|\bar{y}_{ij}(r^{ij}; \underline{\omega}_k)| > |r|$, let $t = 1$ and repeat the following steps until all the $|\bar{y}_{ij}(r^{ij}; \underline{\omega}_k)|$'s are no greater than $|r|$:

(a) Let $t \leftarrow t + 1$.

(b) Select $\underline{\tilde{\omega}}$ (of length $msave$) independent of $\underline{\omega}_k$ and compute $\bar{y}_{ij}(r^{ij}; \underline{\tilde{\omega}})$.

(c) Compute $\bar{y}_{new} = (t-1) \cdot \bar{y}_{ij}(r^{ij}; \underline{\omega}_k)/t + \bar{y}_{ij}(r^{ij}; \underline{\tilde{\omega}})/t$.

(d) Let $\underline{\omega}_k \leftarrow \underline{\omega}_k \cup \underline{\tilde{\omega}}$, $\bar{y}_{ij}(r^{ij}; \underline{\omega}_k) = \bar{y}_{new}$, and $m_k \leftarrow m_k + msave$.

3.4 Find a bounding interval $[l^{ij}, u^{ij}]$ that brackets r_k^{*ij} for $1 \le i < j \le n$:

(a) Initialize $p = 1$, the lower bound $l^{ij} = r^{ij}$, the upper bound $u^{ij} = r^{ij}$, and their function values $\bar{y}_l^{ij} = \rho_{ij} + 1$ and $\bar{y}_u^{ij} = \rho_{ij} - 1$ for $1 \le i < j \le n$.

(b) Update the upper or lower bound:

For $i = 1, ..., n-1$ and $j = i+1, ..., n$:

If $\bar{y}_{ij}(r^{ij}; \underline{\omega}_k) < \rho_{ij}$, update as follows:

If $-1 \le r^{ij} < 1$, set $l^{ij} = r^{ij}$ and $\bar{y}_l^{ij} = \bar{y}_{ij}(r^{ij}; \underline{\omega}_k)$.
Else $(r^{ij} = 1)$, set $r_k^{ij} = 1$ and go to step 4.

If $\bar{y}_{ij}(r^{ij}; \underline{\omega}_k) > \rho_{ij}$, update as follows:

If $-1 < r^{ij} \le 1$, set $u^{ij} = r^{ij}$ and $\bar{y}_u^{ij} = \bar{y}_{ij}(r^{ij}; \underline{\omega}_k)$.
Else $(r^{ij} = -1)$, set $r_k^{ij} = -1$ and go to step 4.

(c) Check bounding intervals: If $(\bar{y}_l^{ij} - \rho_{ij})(\bar{y}_u^{ij} - \rho_{ij}) \le 0$ for all $1 \le i < j \le n$, go to step 3.5. Otherwise, update r^{ij} and δ_k^{ij}, $1 \le i < j \le n$, as follows:

$$r^{ij} \leftarrow \begin{cases} \min(r^{ij} + \delta_k^{ij}, 1) & \text{if } \bar{y}_{ij}(r^{ij}; \underline{\omega}_k) < \rho_{ij} \\ \max(r^{ij} - \delta_k^{ij}, -1) & \text{otherwise} \end{cases}.$$

$$\delta_k^{ij} \leftarrow 2\,\delta_k^{ij}.$$

(d) If $p < maxit$, compute $\bar{y}_{ij}(r^{ij};\underline{\omega}_k)$'s, let $p \leftarrow p+1$ and go to step 3.4 (b). Otherwise, the numerical search fails to find bounds within $maxit$ iterations. In this case, increase the sample size and redo step 3 with the same initial solutions and initial random-number seeds. That is, let $m_k \leftarrow c_1 m_k$ and go to step 3.1.

3.5 Compute the regula-falsi iterate

$$r^{ij} = l^{ij} + (u^{ij} - l^{ij})(\rho_{ij} - \bar{y}_l^{ij})/(\bar{y}_u^{ij} - \bar{y}_l^{ij}) \text{ for } 1 \le i < j \le n.$$

3.6 If $(u^{ij} - l^{ij}) < \epsilon_k$, let $r_k^{ij} = r^{ij}$ and go to step 4. Otherwise, if $p < maxit$, compute $\bar{y}_{ij}(r^{ij};\underline{\omega}_k)$'s, update the upper or lower bound as in step 3.4 (b), let $p \leftarrow p+1$, and go to step 3.5. If $p = maxit$, the numerical search fails to find a solution within error ϵ_k in $maxit$ searches; let $m_k \leftarrow c_1 m_k$ and go to step 3.1.

4. Compute the root estimate $\bar{r}_k^{ij} = \sum_{t=1}^k m_t r_t^{ij} \;/\; \sum_{t=1}^k m_t$ for $1 \le i < j \le n$.

5. Compute $\widehat{\mathrm{Var}}(\bar{r}_k^{ij}) = (\hat{\nu}^{ij})^2 \;/ \sum_{t=1}^k m_t$, where

$$\hat{\nu}^{ij} = \begin{cases} 0 & \text{if } k = 1 \\ \sqrt{(k-1)^{-1}[\sum_{t=1}^k m_t (r_t^{ij})^2 - (\sum_{t=1}^k m_t)(\bar{r}_k^{ij})^2]} & \text{if } k \ge 2 \end{cases}.$$

6. If $k \ge 4$ and $\sqrt{\widehat{\mathrm{Var}}(\bar{r}_k^{ij})} < \sigma_0$ for all $1 \le i < j \le n$, let $\hat{\rho}_{ij}^* = \bar{r}_k^{ij}$ and go to step 7. Otherwise, compute $\epsilon_{k+1} = \epsilon_k/\sqrt{c_1}$, $m_{k+1} = c_1 m_k$, and

$$\delta_{k+1}^{ij} = \begin{cases} \delta_1 & \text{if } k = 1 \\ c_2 \hat{\nu}^{ij} \sqrt{\frac{1}{\sum_{t=1}^k m_t} + \frac{1}{m_{k+1}}} & \text{if } k \ge 2 \end{cases}$$

for $1 \le i < j \le n$, and go to step 1.

7. Let $\hat{\rho}_{ji}^* = \hat{\rho}_{ij}^*$ for $1 \le i < j \le n$ and $\hat{\rho}_{ii} = 1$ for $1 \le i \le n$.

(II) Repeat s times: Apply NORTA to generate one observation of the random vector $\underline{X} = (X_1, ..., X_n)$, using $(\hat{\rho}_{ij}^*)_{i,j=1}^n$ as the correlation matrix for the standard-normal random vector.

Acknowledgments

This research was supported by National Science Council in Taiwan under Grant NSC85-2212-E-212-003. We thank Barry Nelson, Bruce Schmeiser, and Wheyming Tina Song for helpful discussions, Carol Troy for proofreading this paper, and referees for insightful comments.

References

Beasley, J.D., S.G. Springer. 1977. Algorithm AS 111. The percentage points of the normal distribution. *Applied Statistics* **26** 118–121.

Best, D.J., D.E. Roberts. 1975. Algorithm AS 91. The percentage points of the χ^2 distribution. *Applied Statistics* **24** 385–388.

Broste, N.A. 1971. Digital generation of random sequences. *IEEE Transactions on Automatic Control* **AC-16** 213–214.

Cario, M.C., B.L. Nelson. 1996. Autoregressive to anything: time-series input processes for simulation. *Operations Research Letters* **19** 51–58.

Cario, M.C., B.L. Nelson. 1997. Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical Report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.

Cario, M.C., B.L. Nelson. 1998. Numerical methods for fitting and simulating autoregressive-to-anything processes. *INFORMS Journal on Computing* **10** 72–81.

Chen, H. 1994. *Stochastic Root Finding in System Design.* Ph.D. Dissertation, School of Industrial Engineering, Purdue University, West Lafayette, IN.

Chen, H., C. Jeng. 1995. Generation of multivariate random vectors using retrospective approximation algorithms. *Proceedings of the 1995 Chinese Industrial Engineering Conference*, Taoyuan, Taiwan. 839–846.

Chen, H., B.W. Schmeiser. 1994a. Stochastic root finding: problem definition, examples, and algorithms. *Proceedings of the Third Industrial Engineering Research Conference,* Institute of Industrial Engineers, Norcross, GA. 605–610.

Chen, H., B.W. Schmeiser. 1994b. Retrospective approximation algorithms for stochastic root finding. *Proceedings of the 1994 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 255–261.

Chen, H., B.W. Schmeiser. 2001. Stochastic root finding via retrospective approximation. *IIE Transactions* **33** 259–275

Clemen, R.T., G.W. Fischer, R.L. Winkler. 2000. Assessing dependence: some experimental results. *Management Science* **46** 1100–1115.

Clemen, R.T., T. Reilly. 1999. Correlations and copulas for decision and risk analysis.

Management Science **45** 208–224.

Conte, S.D., C. de Boor. 1980. *Elementary Numerical Analysis: An Algorithmic Approach.* McGraw-Hill, New York.

Corner, J.L., C.W. Kirkwood. 1996. The magnitude of errors in proximal multiattribute decision analysis with probabilistically dependent attributes. *Management Science* **42** 1033–1042.

Devroye, L. 1986. *Non-Uniform Random Variate Generation.* Springer Verlag, New York.

Gaver, D.P., P.A.W. Lewis. 1980. First-order autoregressive gamma sequences and point processes. *Adv. Appl. Prob.* **12** 727–745.

Ghosh, S., S. Henderson. 2000. Chessboard distributions and random vectors with specified marginals. Technical Report, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

Gilks, W.R., P. Wild. 1992. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* **41** 337–348.

Gujar, V.G., R.J. Kavanagh. 1968. Generation of random signals with specified probability density functions and power density spectra. *IEEE Transactions on Automatic Control* **AC-13** 716–719.

Hörmann, W. 1995. A rejection technique for sampling from T-concave distributions. *ACM Transactions on Math. Softw.* **21** 182–193.

Huber, P.J. 1964. Robust estimation of a location parameter. *Annals of Mathematical Statistics* **35** 73–101.

Hull, J.C. 1977. Dealing with dependence in risk simulations. *Operational Research Quarterly* **28** 201–213.

Johnson, M.E. 1987. *Multivariate Statistical Simulation.* John Wiley & Sons, New York.

Johnson, M.E., J.S. Ramberg. 1978. Transformation of the multivariate normal distribution with applications to simulation. Technical Report LA-UR-77-2595, Los Alamos Sci. Lab., Los Alamos, NM.

Kendall, M.G., A. Stuart. 1967. *The Advanced Theory of Statistics,* Vol. 1, 3rd ed. Griffin, London, U.K.

Lancaster, H.O. 1957. Some properties of the bivariate normal distribution considered in

the form of a contingency table. *Biometrika* **44** 289–292.

Law, A.M., W.D. Kelton. 2000. *Simulation Modeling and Analysis*, 3rd ed. McGraw-Hill, New York.

Lehmann, E.L. 1983. *Theory of Point Estimation*. John Wiley & Sons, New York.

Lenth, R.V. 1989. Algorithm AS 243. Cumulative distribution function of the noncentral t distribution. *Applied Statistics* **38** 185–189.

Lewis, P.A.W. 1983. Generating negatively correlated gamma variates using the beta-gamma transformation. *Proceedings of the 1983 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 175–176.

Leydold, J. 1998. A rejection technique for sampling from log-concave multivariate distributions. *ACM Transactions on Modeling and Computer Simulation* **8** 254–280.

Li, S.T., J.L. Hammond. 1975. Generation of pseudorandom numbers with specified univariate distributions and correlation coefficients. *IEEE Transactions on System, Man, and Cybernetics* **5** 557–561.

Lurie, P.M., M.S. Goldberg. 1998. An approximate method for sampling correlated random variables from partially specified distributions. *Management Science* **44** 203–218.

Mardia, K.V. 1970. *Families of Bivariate Distributions*. Griffin, London, U.K.

Moonan, W.J. 1957. Linear transformation to a set of stochastically dependent normal variables. *Journal of American Statistical Association* **52** 247–252.

Nelson, B.L., M. Yamnitsky. 1998. Input modeling tools for complex problems. *Proceedings of the 1998 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 105–112.

Parrish, R.S. 1990. Generating random deviates from multivariate Pearson distributions. *Computational Statistics & Data Analysis* **9** 283–295.

Ripley, B.D. 1987. *Stochastic Simulation*. John Wiley & Sons, New York.

Ronning, G. 1977. A simple scheme for generating multivariate gamma distributions with nonnegative covariance matrix. *Technometrics* **19** 179–183.

Scheuer, E.M., D.S. Stoller. 1962. On the generation of normal random vectors. *Technometrics* **4** 278–281.

Schmeiser, B.W. 1990. Simulation experiments. D.P. Heyman, M.J. Sobel, eds. *Handbooks*

in Operations Research and Management Science **2**. North-Holland, New York. 295–330.

Schmeiser, B.W. 1999. Advanced input modeling for simulation experimentation. *Proceedings of the 1999 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 110–115.

Schmeiser, B.W., R. Lal. 1982. Bivariate gamma random vectors. *Operations Research* **30** 355–374.

Song, W.T., L. Hsiao. 1993. Generation of autocorrelated random variables with a specified marginal distribution. *Proceedings of the 1993 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 374–377.

Song, W.T., L. Hsiao, Y. Chen. 1996. Generating pseudo-random time series with specified marginal distributions. *European Journal of Operational Research* **94** 194–202.

Stanfield, P.M., J.R. Wilson, G.A. Mirka, N.F. Glasscock, J.P. Psihogios, J.R. Davis. 1996. Multivariate input modeling with Johnson distributions. *Proceedings of the 1996 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 1457–1464.

Accepted by Michael R. Taaffe; received April 1999; revised April 2000, January 2001; accepted April 2001.