# Computing Multivariate Normal Probabilities: A New Look

H. I. GASSMANN, I. DEÁK, and T. SZÁNTAI

This article describes and compares several numerical methods for finding multivariate probabilities over a rectangle. A large computational study shows how the computation times depend on the problem dimensions, the correlation structure, the magnitude of the sought probability, and the required accuracy. No method is uniformly best for all problems and—unlike previous work—this article gives some guidelines to help establish the most promising method a priori. Numerical tests were conducted on approximately 3,000 problems generated randomly in up to 20 dimensions. Our findings indicate that direct integration methods give acceptable results for up to 12-dimensional problems, provided that the probability mass of the rectangle is not too large (less than about 0.9). For problems with small probabilities (less than 0.3) a crude Monte Carlo method gives reasonable results quickly, while bounding procedures perform best on problems with large probabilities ($> 0.9$). For larger problems numerical integration with quasirandom Korobov points may be considered, as may a decomposition method due to Deák. The best method found four-digit accurate probabilities for every 20-dimensional problem in less than six minutes on a 533MHz Pentium III computer.

**Key Words:** Computation; Multivariate normal distribution; Rectangular probability.



## 1. INTRODUCTION

The problem considered in this article is that of finding the probability of a multivariate normal random variable over a rectangle. We begin with a formal definition of the problem.

Let an $n$-dimensional rectangle $(\mathbf{a}, \mathbf{b}) = (a_1, b_1) \times (a_2, b_2) \times \cdots \times (a_n, b_n)$ and a

H. I. Gassmann is Associate Professor, School of Business Administration, Dalhousie University, 6152 Coburg Road, Halifax Nova Scotia, B3H 3J4, Canada (E-mail: hgassman@mgmt.dal.ca). I. Deák is Professor, and T. Szántai is Professor, Department of Differential Equations, Budapest University of Technology and Economics, Muegyetem Rakpart 3, H-1521 Budapest, Hungary (E-mail addresses: deak@math.bme.hu; szantai@math.bme.hu).



*Journal of Computational and Graphical Statistics, Volume 11, Number 4, Pages 920–949*
*DOI: 10.1198/106186002385*

symmetric positive definite matrix $\Sigma$ be given. The problem is then to find

$$p = P(\mathbf{a}, \mathbf{b}) = \int_{a_1}^{b_1} \cdots \int_{a_n}^{b_n} \frac{1}{|\Sigma|^{1/2}(2\pi)^{n/2}} e^{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)} d\mathbf{x}, \tag{1.1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$. Some components of $\mathbf{a}$ and $\mathbf{b}$ may be infinite, and if $\mathbf{a} = (-\infty, \ldots, -\infty)$, then $P(\mathbf{a}, \mathbf{b})$ is the value of the distribution function $F(\mathbf{b})$. We do assume that for each $i$ at least one of $a_i$ and $b_i$ is finite. (If $(a_i, b_i) = (-\infty, +\infty)$, then a suitable permutation of components allows the $i$th coordinate to be integrated out explicitly, reducing the overall problem dimension.) We assume further that $\Sigma$ is a correlation matrix, that is, $\text{diag}(\Sigma) = (1, 1, \ldots, 1)$ and that $\mu = 0$. This does not constitute a loss of generality, since it is easy to standardize each component by the transformation $y_i = (x_i - \mu_i)/\sqrt{\sigma_{ii}}$.

The problem has received considerable attention in the literature, including theoretical papers by Kendall (1941), Gupta (1963), and Harris and Soms (1980). Many different numerical methods have been proposed (see, e.g., Deák 1980; Schervish 1984; Szántai 1976, 2000; Genz 1992, 1993; Gassmann 1988; Beckers and Haegemans 1992; Joe 1995; Hajivassiliou, McFadden, and Ruud 1996; Somerville 1998 and Vijverberg 1997).

Statistical applications include the multivariate probit model (e.g., Ochi and Prentice 1984), the multivariate ordinal response model (e.g., Anderson and Pemberton 1985) and multivariate paired comparisons (Böckenholt 1992). There are also applications in stochastic programming (see e.g., Prékopa 1974, 1995), water management (Prékopa, Deák, Ganczer, and Patyi 1980) and energy management (Prékopa and Szántai 1978).

There are two reasons for writing this article. On one hand, the recent article by Szántai (2000) developed several new bounding techniques whose computational qualities have not yet been fully explored. On the other hand, previous computational studies have tended to try to establish global superiority of one method over others, without regard to specifics of the problem such as the nature of the correlation matrix and the probability $p$. This article will specifically address the question of how characteristics such as these will affect the efficiency of the methods tested.

The outline of the article is as follows: Section 2 describes several methods based on multivariate integration techniques. Ordinary Monte Carlo methods are treated in Section 3 along with some common variance reduction techniques. These include a method introduced by Deák (1980, 1990), which decomposes the multivariate normal distribution into two components, an $(n-1)$-dimensional direction and a one-dimensional length component, which can be integrated out explicitly. This method was recently extended by Deák (in press) to allow computation of multivariate normal probabilities over general sets defined as the intersection of a finite number of linear and quadratic inequalities. Another group of algorithms due to Szántai is based on computing upper and lower bounds on the probability $p$. These methods are described in Section 4. Section 5 contains comparative numerical results; some extensions and concluding remarks are contained in Section 6.

## 2. INTEGRATION METHODS

Equation (1.1) is an ordinary (multidimensional) integral, which can be solved by standard integration techniques, at least in principle. For one- and two-dimensional methods various efficient methods exist (see, e.g., Hill and Joyce 1967, Hart et al. 1968; Donnelly 1973; Drezner and Wesolowsky 1990). Another easy case occurs when the components of $\mathbf{x}$ are equicorrelated. If $\sigma_{ij} = \rho$ for all $i \neq j$, with $\rho \geq 0$, then (Tong 1990, p. 192) $P(\mathbf{a}, \mathbf{b})$ can be found by the formula

$$P(\mathbf{a}, \mathbf{b}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}t^2} \prod_{i=1}^{n} \left[ \Phi((b_i + \sqrt{\rho}t)/\sqrt{1-\rho}) - \Phi((a_i + \sqrt{\rho}t)/\sqrt{1-\rho}) \right] dt,$$

which is essentially a two-dimensional integral. If $\rho = 0$, further simplifications are possible.



### 2.1 SCHERVISH'S METHOD

As described by Schervish (1984), this method is a straightforward application of nested Newton–Cotes quadrature rules over the integration region $(\mathbf{a}, \mathbf{b})$. The components of $\mathbf{x}$ are reordered so that the two largest integration regions correspond to the innermost integrals, which are then evaluated using the highly accurate algorithm by Donnelly (1973). Donnelly's algorithm explicitly provides for infinite integration bounds of the innermost components; otherwise infinite regions are truncated sufficiently far in the tail of the distribution.

One of the features of Schervish's algorithm is that it has error bounds that are deterministic, unlike the statistical error bounds of some methods described later.

### 2.2 GENZ'S METHOD

The main idea is to transform the integration region to the unit cube $[0, 1]^n$ by a sequence of elementary transformations. This comes at the expense of a slightly more complicated integrand.

The sequence begins with the Cholesky transformation $\mathbf{x} = C\mathbf{y}$, where $C$ is the Cholesky factor of $\Sigma$, that is, $C$ is lower triangular and $\Sigma = CC'$. The components of $\mathbf{y}$ are univariate standard normal random variables and are independent of each other.

One further computes $\mathbf{x}'\Sigma^{-1}\mathbf{x} = (\mathbf{y}'C')((C')^{-1}C^{-1})(C\mathbf{y}) = \mathbf{y}'\mathbf{y}$ and $d\mathbf{x} = |C|d\mathbf{y} = |\Sigma^{1/2}|d\mathbf{y}$. Moreover, the inequality $\mathbf{a} \leq \mathbf{x} = C\mathbf{y} \leq \mathbf{b}$ decomposes into the following set of inequalities:

$$\begin{aligned} a_1 &\leq y_1 \leq b_1 \\ \left( a_i - \sum_{j<i} c_{ij} y_j \right) / c_{ii} &\leq y_i \leq \left( b_i - \sum_{j<i} c_{ij} y_j \right) / c_{ii}, \quad i = 2, \ldots, n. \end{aligned}$$

The integral (1.1) is therefore transformed into the iterated integral

$$p = \int_{a_1}^{b_1} \frac{1}{\sqrt{2\pi}} e^{-y_1^2/2} \int_{a_2'(y_1)}^{b_2'(y_1)} \frac{1}{\sqrt{2\pi}} e^{-y_2^2/2} \cdots \int_{a_n'(y_1,\ldots,y_{n-1})}^{b_n'(y_1,\ldots,y_{n-1})} \frac{1}{\sqrt{2\pi}} e^{-y_n^2/2} dy_n, \ldots dy_1,$$

where

$$a_i'(y_1, \ldots, y_{i-1}) = \left( a_i - \sum_{j<i} c_{ij} y_j \right) / c_{ii},$$

and

$$b_i'(y_1, \ldots, y_{i-1}) = \left( b_i - \sum_{j<i} c_{ij} y_j \right) / c_{ii}.$$

Next each $y_i$ is transformed separately using $y_i = \Phi^{-1}(z_i)$, where $\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$. Now

$$p = \int_{g_1}^{h_1} \int_{g_2(z_1)}^{h_2(z_1)} \cdots \int_{g_n(z_1,z_2,\ldots,z_{n-1})}^{g_n(z_1,z_2,\ldots,z_{n-1})} dz_n \ldots dz_1$$

where one defines $g_1 = \Phi(a_1)$, $h_1 = \Phi(b_1)$, $g_2(z_1) = \Phi((a_2 - c_{12}\Phi^{-1}(z_1))/c_{11})$, $h_2(z_1) = \Phi((b_2 - c_{12}\Phi^{-1}(z_1))/c_{11})$, and for general $i = 3, \ldots, n$, $g_i(z_1, \ldots, z_{i-1}) = \Phi((a_i - \sum_{j=1}^{i-1} c_{ij}\Phi^{-1}(z_j))/c_{ii})$, and $h_i(z_1, \ldots, z_{i-1}) = \Phi((b_i - \sum_{j=1}^{i-1} c_{ij}\Phi^{-1}(z_j))/c_{ii})$.

Finally, the transformation $z_i = g_i + w_i(h_i - g_i)$ standardizes the integration region, giving

$$p = (h_1 - g_1) \int_0^1 (h_2(w_1) - g_2(w_1)) \int_0^1 \cdots \int_0^1 (h_n(w_1, \ldots, w_{n-1}) - g_n(w_1, \ldots, w_{n-1})) dw_{n-1} \ldots dw_1,$$

since each integrand $f_i = h_i - g_i$ is independent of $w_i$ and the innermost integral $\int_0^1 dw_n = 1$ can be ignored.

This sequence of transformations has also forced a priority ordering on the components of $\mathbf{x}$ which makes the problem amenable to the application of subregion adaptive algorithms. The method works best if the components are presorted so that the innermost integration has the most "weight." This can be done statically (i.e., by sorting the components once in increasing order of $\Phi(b_i) - \Phi(a_i)$) or dynamically.

Genz describes three different methods for solving this transformed integral. His first method is based on a polynomial approximation of the integrand. For better performance, the unit cube is split into subregions which are subsequently partitioned further whenever the approximation is not accurate enough. Explicit error bounds are available for each subregion and can be aggregated into an overall error bound.

A second method uses quasi-random integration points. This method was originally described by Korobov (1959, 1960) and Saltykov (1963) and subsequently refined by

Keast (1973). The third method uses pseudo-random integration points. Here the error estimates are statistical in nature, as in the simulation methods of the next section.

## 3. SIMULATION METHODS

This section looks at three methods which attempt to solve the problem by statistical means. These methods are collectively known as "Monte Carlo methods."

### 3.1 CRUDE MONTE CARLO

The simplest Monte Carlo method is also the most widely used. Simply generate a large number $N$ of points whose distribution follows the normal $(0, \Sigma)$ law. Record the number $K$ of these points that fall inside the rectangle $(\mathbf{a}, \mathbf{b})$ and return the ratio $\widehat{p} = K/N$ as an estimate of the value $p$. Since the points were randomly generated, $\widehat{p}$ is a random variable, and performance guarantees are statistical. It obtains that $\mathrm{E}(\widehat{p}) = p$, $\mathrm{var}(\widehat{p}) = p(1-p)/N$, and an approximate 95% confidence interval of $p$ is given by $\widehat{p} \pm 1.96 * \sqrt{\widehat{p}(1-\widehat{p})/N}$. The square root is maximized when $\widehat{p} = 0.5$, so when $N = 100$ we can expect (with 95% confidence) to obtain an estimate of any $p$ to within at most 0.1. For higher accuracy, a larger sample size is required. In particular, if we want to halve the width of the confidence interval, we must quadruple the number of points in the sample.

How should the points be generated? The easiest way is to generate $n$ univariate standard normal variates $z_i$, for example, by the polar method (Deák 1990), and to compute the Cholesky transformation $\mathbf{x} = C\mathbf{z}$.

The method can be speeded up somewhat by judicious reordering of the components of $\mathbf{x}$. Typically the components are checked sequentially for satisfying the constraints $a_i \leq x_i \leq b_i$. Whenever one constraint is violated, one can quit checking, so it is useful to reorder the components so as to break from the loop as soon as possible, that is, the components of $\mathbf{x}$ should again be ordered in increasing order of $\Phi(b_i) - \Phi(a_i)$. Further savings are possible if each component of $\mathbf{x}$ is computed only *after* its predecessors have been checked.

### 3.2 VARIANCE REDUCTION TECHNIQUES

There are several ways to reduce the variance of the estimator $\widehat{p}$ in the preceding section. The easiest way is to use antithetic random variables, that is, whenever a point $\mathbf{x}^k$ is generated, one also uses the point $-\mathbf{x}^k$. Since points in this method are generated in pairs, it makes sense to treat both of them together and to write the estimator in the form $\widehat{p} = \frac{1}{N} \sum_{k=1}^{N} f(\mathbf{x}^k)$ where

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if both } \mathbf{x} \text{ and } -\mathbf{x} \text{ lie in } (\mathbf{a}, \mathbf{b}) \\ 1/2 & \text{if exactly one of } \mathbf{x} \text{ and } -\mathbf{x} \text{ lie in } (\mathbf{a}, \mathbf{b}) \\ 0 & \text{if both } \mathbf{x} \text{ and } -\mathbf{x} \text{ lie outside of } (\mathbf{a}, \mathbf{b}). \end{cases}$$

Letting $p_2$ be the probability that both $\mathbf{x}^k$ and $-\mathbf{x}^k$ are contained in $(\mathbf{a}, \mathbf{b})$, it is an elementary exercise to show that the variance of this estimator is $\left[p(1-p) - 1/2(p-p_2)\right]/N$. Since $p_2 \leq p$, the variance is never larger than in the simple Monte Carlo method, although nothing is gained if the interval $(\mathbf{a}, \mathbf{b})$ is symmetric about the origin.

More complicated variance reduction schemes are possible and will be discussed in the next section.

### 3.3 DEÁK'S DECOMPOSITION

The main thrust of this approach is to decompose the normal random variable $\mathbf{x}$ into two parts, a direction and a distance from the origin. This decomposition can be used both in the generation of sample points $\mathbf{x}^k$ and in the calculation of the probability $p$.

Following Deák (1980), we write $\mathbf{x} = C\lambda\mathbf{s}$, where $C$ is the Cholesky factor of $\Sigma$, $\lambda$ has a chi-distribution with $n$ degrees of freedom, and $\mathbf{s}$ is uniformly distributed on the $n$-dimensional unit sphere, independently of $\lambda$. We can now rewrite (1.1) in the following form:

$$\begin{aligned} p &= \int_{\mathrm{R}^n} \mathbf{1}_{(\mathbf{a},\mathbf{b})}(\mathbf{x}) dF(\mathbf{x}) \\ &= \int_{S^n} \int_{(0,\infty)} \mathbf{1}_{(\mathbf{a},\mathbf{b})}(C\lambda\mathbf{s}) d\chi_n(\lambda) dU(\mathbf{s}) \\ &= \int_{S^n} \int_{r_1(\mathbf{s})}^{r_2(\mathbf{s})} d\chi_n(\lambda) dU(\mathbf{s}), \end{aligned}$$

where

$$\begin{aligned} \mathbf{1}_{(\mathbf{a},\mathbf{b})}(\mathbf{x}) &= \begin{cases} 1 & \text{if } \mathbf{x} \in (\mathbf{a}, \mathbf{b}) \\ 0 & \text{otherwise,} \end{cases} \\ r_1(\mathbf{s}) &= \min\{r : r \geq 0, \mathbf{a} \leq rC\mathbf{s} \leq \mathbf{b}\}, \\ r_2(\mathbf{s}) &= \max\{r : r \geq 0, \mathbf{a} \leq rC\mathbf{s} \leq \mathbf{b}\}. \end{aligned}$$

The inner one-dimensional integral can be handled by numerical integration (based on the chi-squared distribution function), the outer integral is approximated by sampling techniques. A simple Monte Carlo method might generate $N$ sample points $\mathbf{s}^k$ uniformly distributed on the unit sphere $S^n$ and form the estimator

$$\widehat{p} = \frac{1}{N} \sum_{k=1}^{N} f(\mathbf{s}^k), \quad \text{where} \quad f(\mathbf{s}) = \int_{r_1(\mathbf{s})}^{r_2(\mathbf{s})} d\chi_n(\lambda),$$

or one could use antithetic random variables to reduce the variance. Deák devised an improvement over this scheme that is intended to distribute a large number of directions as uniformly as possible on the unit sphere.

A set of $n$ directions is chosen first and converted into an orthonormal system, that is, $n$ unit vectors that are mutually orthogonal. From each orthonormal system $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$

one obtains $2^m\binom{n}{m}$ directions by computing the sum

$$d(v, l_1, \ldots, l_m) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} v_j \mathbf{s}_{l_j},$$

for every sign vector $v = (v_1, \ldots, v_m)$ (each component is either $+1$ or $-1$), and every combination of indices satisfying $1 \le l_1 < \ldots < l_m \le n$.

An estimator for $p$ can then be written as

$$\widehat{p} = \frac{1}{N} \sum_{k=1}^{N} f(\mathbf{s}_1^k, \ldots, \mathbf{s}_n^k), \tag{3.1}$$

where $(\mathbf{s}_1^k, \ldots, \mathbf{s}_n^k)$ is an orthonormal system and

$$f(\mathbf{s}_1^k, \ldots, \mathbf{s}_n^k) = \frac{1}{2^m \binom{n}{m}} \sum_{v} \sum_{l_1 < \ldots < l_m} \int_{r_1(d(v,l_1,\ldots,l_m))}^{r_2(d(v,l_1,\ldots,l_m))} d\chi_n(\lambda).$$

The parameter $m$ can in principle be chosen arbitrarily from the set $\{1, 2, \ldots, n\}$, but the computational complexity increases very fast. Best results are obtained for $m = 2$ or $m = 3$.

It is clear that the summand in formula (3.1) always lies between 0 and 1, and so the variance of even the simplest Deák estimator is less than the variance of the crude Monte Carlo method, for a given sample size $N$.

## 4. BOUNDING METHODS

This class of methods seeks to develop upper and lower bounds for the probability $p$, obtained from a suitable combination of lower dimensional marginal probabilities, which are easy to compute. The most basic of these bounds are so-called Bonferroni bounds, which are obtained as an easy consequence of de Morgan's rules.

### 4.1 SZÁNTAI'S APPROACH

In a number of articles Szántai (1976, 1986, 2000) has developed several computational methods for the computation of multivariate distribution functions. The procedure is applicable to a variety of distributions, including the multivariate normal distribution and can be adapted easily to calculating rectangular probabilities.

We first rewrite the rectangle $(\mathbf{a}, \mathbf{b})$ as an intersection of half-spaces. (Each finite component of $\mathbf{a}$ and $\mathbf{b}$ defines one such half-space through the restriction $x_i \ge a_i$ or $x_i \le b_i$, respectively.) By suitably pairing up half-spaces when $a_i$ and $b_i$ are both finite for some $i$, one arrives at the decomposition

$$A := (\mathbf{a}, \mathbf{b}) = \bigcap_{i=1}^{n} A_i, \quad \text{where} \quad A_i = \{\mathbf{a} \in R^n : a_i \le x_i \le b_i\}.$$

Writing $\bar{A}_i = R^n \setminus A_i$, one then obtains the following alternating expansion for $P(A)$:

$$P(A) = 1 - \sum_{i=1}^{n} P(\bar{A}_i) + \sum_{i<j} P(\bar{A}_i \cap \bar{A}_j) - \sum_{i<j<k} P(\bar{A}_i \cap \bar{A}_j \cap \bar{A}_k) + - \ldots + (-1)^n P\left(\bigcap_{i=1}^{n} \bar{A}_i\right).$$

Truncating this expansion after $k$ terms gives a lower or upper bound depending on whether $k$ is even or odd, respectively.

For instance,

$$B_1 := 1 - \sum_{i=1}^{n} P(\bar{A}_i) \le P(A) \le 1 - \sum_{i=1}^{n} P(\bar{A}_i) + \sum_{i<j} P(\bar{A}_i \cap \bar{A}_j) =: B_2. \qquad (4.1)$$

Both of these bounds can be computed explicitly, since they involve only one- and two-dimensional normal marginal probabilities.

Other bounds were developed by Hunter (1976), Worsley (1982), Tomescu (1986), Bukszar and Szántai (1999), Bukszar and Prékopa (2001), using one-, two-, and three-dimensional marginals. Szántai (1986) shows how the bounds can be used to compute three separate estimators for $p$ from the same sample by

i. sampling $p$ directly using simple Monte Carlo sampling,
ii. sampling for the difference between $p$ and a lower bound $p_L$,
iii. sampling for the difference between $p$ and an upper bound $p_U$.

This gives three separate (unbiased) estimators for $p$, which we denote by $\widehat{p}_1, \widehat{p}_2 := \widehat{p - p_L} + p_L$ and $\widehat{p}_3 := \widehat{p - p_U} + p_U$. Each estimator can be found by simple Monte Carlo sampling. If for instance $p_L = B_1$, the lower bound in Equation (4.1), then

$$\widehat{p - p_L} + p_L = B_1 + \frac{1}{N} \sum_{k=1}^{N} f(\mathbf{x}^k),$$

where

$$f(\mathbf{x}) = \begin{cases} 0 & \text{if} \quad \mathbf{x} \in (\mathbf{a}, \mathbf{b}) \\ j - 1 & \text{if} \quad \mathbf{x} \text{ is contained in exactly } j \text{ of the sets } \bar{A}_1, \ldots, \bar{A}_n. \end{cases}$$

Similar results hold for all other bounds, and the summand in each case depends only on the number of half-spaces that contain (or fail to contain) the sampled point $\mathbf{x}^i$. The estimators tend to be negatively correlated, and they can be combined by finding the linear combination of least variance, that is, one can form

$$\widehat{p}_{\text{final}} = w_1 \widehat{p}_1 + w_2 \widehat{p}_2 + w_3 \widehat{p}_3, \quad \text{where} \quad w_1 + w_2 + w_3 = 1.$$

This implicit minimization problem can be solved analytically given the sample variances and covariances of $\widehat{p}_1$, $\widehat{p}_2$ and $\widehat{p}_3$ (see Prékopa 1995). Since the variance of $\widehat{p}_{\text{final}}$ can never be larger than $\text{var}(\widehat{p}_1)$, this method can also be thought of as a scheme to reduce the variance of the crude Monte Carlo method.

### 4.2 A HYBRID METHOD

This section shows how the methods of Szántai and Deák can be combined into a class of algorithms that combine the advantages of both (see also Gassmann 1988).

Monte Carlo methods, Deák's decomposition and Szántai's estimators all have in common that they randomly sample objects and form the average of a *weight function.* Differences between the methods concern the objects that are sampled and the weight function used.

In the crude Monte Carlo method, the objects are points in $R^n$ and the weight function is the indicator function $\mathbf{1}_{(\mathbf{a},\mathbf{b})}$. In Deák's method, the objects are rays and the weight function for each direction $\mathbf{s}$ is the integral

$$\int_0^\infty \mathbf{1}_{(\mathbf{a},\mathbf{b})}(\lambda C\mathbf{s})d\chi_n(\lambda).$$

Szántai uses points as the sampled objects, together with a variety of weight functions, depending on which bound is used. The crucial observation is that to every weight function $w(\mathbf{x})$ used in Szántai's method for some estimator $\widehat{p}_S$, there is a corresponding hybrid estimator $\widehat{p}_H$ using the integrated weight function

$$\int_0^\infty w(\lambda C\mathbf{s})d\chi_n(\lambda) \tag{4.2}$$

with rays instead of points. The integrand in Equation (4.2) is in fact a step function, which can be evaluated by picking one representative point in each of the regions defined by the half-spaces that make up $(\mathbf{a}, \mathbf{b})$.

The hybrid method can thus be summarized as follows:

1. Pick $N$ and for each $i = 1, \ldots, N$ form an orthonormal system $\{s_1, \ldots, s_n\}$ as in Deák's method.
2. Form $2^m \binom{n}{m}$ linear combinations $d(v, l_1, \ldots, l_m)$ of these vectors using $m$ orthonormal vectors at a time together with a sign vector $v$.
3. For each linear combination determine the points of intersection with the hyperplanes that make up the rectangle $(\mathbf{a}, \mathbf{b})$. If we write $d_1, \ldots, d_n$ for the components of the vector $d(v, l_1, \ldots, l_m)$, then the points of intersection are $\lambda_i = a_i/d_i$ and $\lambda_{n+i} = b_i/d_i$. (With probability 1 $d_i$ is nonzero.)
4. Ignore negative and infinite $\lambda_i$, sort the remaining ones in increasing order and renumber them, so that $0 \le \lambda_1 \le \cdots \le \lambda_K$. Set $\lambda_{K+1} = +\infty$.
5. For each $k = 1, \ldots, K$ compute the integral $I_k = \int_{\lambda_k}^{\lambda_{k+1}} d\chi_n(\lambda)$ and find the number $j$ of inequalities that hold at the midpoint $c := (\lambda_k + \lambda_{k+1})/2$ $d(v, l_1, \ldots, l_m)$.
6. Determine the weights $t_1^S$, $t_2^S$, and $t_3^S$ with which the midpoint $c$ would impact on Szántai's estimators. (This will depend on $j$ and on the bound used. For instance, $t_1^S = \mathbf{1}_{(\mathbf{a},\mathbf{b})}(c)$ and $t_2^S = \max\{0, j-1\}$ if the lower bound used is $B_1$.) Add $t_i^S * I_k/(2^m \binom{n}{m})$ to the hybrid estimators $\widehat{p}_i^H, i = 1, 2, 3$.
7. Determine the weights $w_1, w_2$ and $w_3$ that minimize the variance of the final estimator $\widehat{p}_{\text{final}}^H = w_1\widehat{p}_1^H + w_2\widehat{p}_2^H + w_3\widehat{p}_3^H$. Return $\widehat{p}_{\text{final}}^H$ as the estimate of the probability $p$.

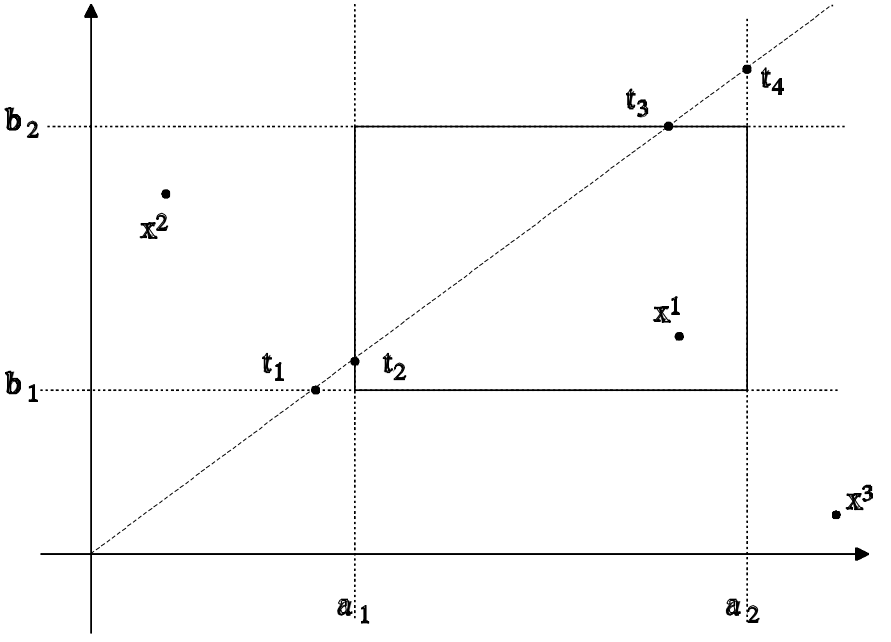


*Figure 1. Illustration of the hybrid method.*

A two-dimensional example may serve as an illustration (see Figure 1). Crude Monte Carlo gives weight 1 to point $\mathbf{x}^1$ and weight 0 to $\mathbf{x}^2$ and $\mathbf{x}^3$. Deák's method integrates the $\chi$-distribution over the interval $[t_2, t_3]$ and ignores the rest of the ray. Szántai's method squeezes some further information out of the points $\mathbf{x}^2$ and $\mathbf{x}^3$: $\mathbf{x}^3$ will be given weight 1 in the estimator $\widehat{p_{B_1}}$ (or $\widehat{p}_2^S$). The hybrid method contributes $\int_{t_2}^{t_3} d\chi_n(\lambda)$ to the estimator $\widehat{p}_1^H$ and $\int_0^{t_1} d\chi_n(\lambda) + \int_{t_4}^{\infty} d\chi_n(\lambda)$ to $\widehat{p}_2^H$.

## 5. NUMERICAL TESTS

This section presents a large number of tests designed to study the behavior of all the methods described so far. We are especially interested in formulating a priori rules to select the most promising method, so we will try to identify all factors that might influence computation times and precision.

The first and most obvious factor is the problem dimension $n$. Generally speaking, as $n$ increases, we expect the computation times to increase. Different methods may follow different patterns, and it will be interesting to know their limitations.

The second factor is the required accuracy. The more accuracy one requires, the longer it will take to complete the computations. But the overhead for different methods varies, so the accuracy will affect these methods differently.

Third, the position and shape of the rectangle may influence computation times. For instance, if the rectangle is symmetric about the origin, antithetic variables will be less effective than for rectangles that do not contain the origin. Deák's method may also be

affected, since rays pointing in different directions may have markedly different characteristics. Moreover, rectangles that are very long and thin may behave differently from rectangles whose sides are more nearly equal to each other.

A fourth factor is the value of $p$, which affects the variance of the Monte Carlo estimators. If $p$ is very large or very small, substantially fewer points will have to be generated to achieve a preset accuracy requirement, so the computation times should be decreased as well. (Of course, it is hard to assess $p$ without actually solving the problem, but some preliminary estimates may be computable without too much effort.)

The final factor considered in this study is the composition of the correlation matrix. Does it have many large positive correlations, for example, or a preponderance of negative entries, or any other easily discernible pattern?

All in all, nine different methods were considered in the numerical tests:

1. A crude Monte Carlo method applied to Genz's transformed problem (RANMVN);
2. Genz's subregion adaptive partitioning algorithm (SADMVN);
3. Genz's implementation of randomized Korobov rules (KROMVN);
4. Schervish's integration method (MULNOR);
5. Deák's orthonormal variates (with $m = 2$) (DE2MVN);
6. Szántai's bounding procedure (SZ3MVN);
7. the hybrid approach described in Section 4.2 (HYBRID);
8. Szántai's method using only two-dimensional information to compute the bounds (SZ2MVN);
9. the hybrid method with two-dimensional information (HYB2).

Genz's implementations (RANMVN, SADMVN, KROMVN) were downloaded from his Web site (Genz 1998), and M. Schervish graciously made available his code MULNOR; the other implementations are the authors'.

For different problem sizes ranging from $n = 3$ to $n = 20$, a large number of problems were computed to varying degrees of accuracy. Several classes of correlation matrices were generated randomly, using methods described by Marsaglia and Olkin (1984): perturbations of three equicorrelated matrices (with correlation coefficients of 0, 0.95, and $-1/n$—the last since the correlation coefficient $\rho$ in an equicorrelated matrix must be greater than $-1/(n-1)$), and a general correlation matrix with no further restrictions. Five matrices were generated in each class. For each matrix, two small rectangles were randomly generated, one containing zero, the other contained entirely within the positive orthant. Each rectangle was then systematically enlarged in several steps until the probability exceeded 0.99. (This required between six and nine steps.)

The accuracy requirements were as follows. The integration methods (SADMVN, MULNOR) were stopped when the absolute accuracy was reported less than 0.01, 0.001, and 0.0001. All other methods were stopped when the standard deviation of the error was less than one third of those figures. Problems with very small probabilities ($< 0.01$) were omitted so as to sidestep the issue of absolute versus relative accuracy.

Table 1 shows cumulative times over all problems to give an idea of how the different methods compare and how they behave as the dimension $n$ increases. Each figure reported

is the sum of the computation times for about 250 problems. (The exact number varies with the problem dimension $n$ as indicated in the third column of the table.) The most dramatic increase in computation times occurs for Schervish's method, while Szántai's method and Deák's orthonormal variates are at the other end of the spectrum. This table does not tell the whole story, however, because within each method there are tremendous differences from one problem to the next, and it is essential to take into account the special features of each problem, in particular the probability $p$ and the desired accuracy. These relationships will be explored next.

In order not to overwhelm the reader, we first study three- and four-dimensional problems and draw some conclusions, using them to cut down on the possibilities considered. Spot checks will verify the validity of the conclusions for problems in higher dimensions.



### 5.1 THE CASE $n = 3$

This series of tests comprises 252 separate problems whose probability was estimated using seven of the methods. (Szántai's method SZ3MVN and the hybrid method were excluded since they require computation of three-dimensional marginals by some other means.)

Figure 2 shows the computation times of the seven methods. (All procedures were compiled with Lahey Fortran 90 and executed on a 533 MHz Pentium III computer with 128 Mb of RAM.) This particular series of problems uses a tolerance limit of 0.0001. Other instances with less stringent requirements were also run, but their times are so much shorter that accurate reporting is not possible for the fastest routines (SADMVN, MULNOR), since the resolution of the clock is only 0.01 seconds.

Several facts are apparent from the graph. First, the subregion adaptive method (SADMVN) takes significantly less computation time than any other method, and only Schervish's code might be considered as a possible alternative. The superior performance of SADMVN persists over the entire range of problems considered. The hybrid method with two-dimensional marginals is slightly better than Szántai's method. Finally it is interesting to observe the behavior of the various methods as the size of the integration region is increased. The sampling methods (RANMVN, DE2MVN) take longest when the probability is near 0.5 (and the variance of the estimators is largest). The time taken by the integration methods (SADMVN, KROMVN, MULNOR) increases (superlinearly) as the probability increases. Most interesting are the bounding methods, where the computation time *decreases* with increasing $p$. This is consistent with the fact that the bounds tend to get tighter as the probability increases. When the probability exceeds 0.9 it is not uncommon to see the upper and lower bounds differing by less than $10^{-6}$, so the most time consuming part of these methods, the sampling phase, can be dispensed with altogether.

It is hard to reach conclusions concerning the dependence of computation times on the location of the rectangle or the type of correlation matrix, although SADMVN seems to take slightly longer for rectangles not containing the origin. The nearly independent problems are solved by all methods in significantly less time than the other problems.

Table 1. Aggregate Times for all Methods and Varying Dimensions *n*

| | | | Method | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | Dim *n* | N of problems | Crude MC | SADMVN | Korobov | Schervish | Ortho 2 | Szántai | Hybrid | Szántai 2D | Hybrid 2D |
| 0.0001 | 3 | 252 | 4171.86 | 0.25 | 7.97 | 0.67 | 2963.67 | NA | NA | 3422.37 | 1629.19 |
| | 4 | 247 | 6312.50 | 0.99 | 34.10 | 91.70 | 2624.50 | 1185.10 | 497.89 | 7358.51 | 3611.74 |
| | 5 | 243 | 13618.92 | 4.97 | 84.54 | 23424.98 | 3869.74 | 3085.31 | 1601.78 | 10717.22 | 7051.67 |
| | 6 | 238 | 23584.57 | 22.41 | 291.15 | NA | 4245.30 | 4347.27 | 2329.81 | 16351.87 | 10916.86 |
| | 7 | 233 | 27874.40 | 76.66 | 578.94 | NA | 7426.52 | 6594.94 | 5176.79 | 21528.55 | 20370.69 |
| | 8 | 232 | 33279.52 | 197.83 | 835.64 | NA | 7955.94 | 10510.97 | 9262.29 | 28062.57 | 26773.96 |
| | 10 | 226 | 50902.95 | 1141.46 | 3535.34 | NA | 12055.48 | 10674.88 | 17048.23 | 35307.91 | 58572.63 |
| | 12 | 224 | 64477.00 | 15230.24 | 32066.99 | NA | 15003.29 | 13003.05 | 31034.69 | 41016.82 | 97919.55 |
| | 15 | 216 | 95604.34 | 163065.20 | 14499.37 | NA | 28952.15 | NA | NA | 54206.47 | 219894.50 |
| | 20 | 220 | 139694.40 | 311885.50 | 61708.67 | NA | 31336.11 | NA | NA | 82763.27 | NA |
| 0.001 | 3 | 252 | 43.31 | 0.06 | 1.96 | 0.41 | 31.68 | NA | NA | 34.71 | 16.65 |
| | 4 | 247 | 85.91 | 0.19 | 3.97 | 45.39 | 25.95 | 17.16 | 12.11 | 73.45 | 34.52 |
| | 5 | 243 | 136.42 | 0.84 | 5.94 | 8594.07 | 44.43 | 44.82 | 36.30 | 109.97 | 74.92 |
| | 6 | 238 | 242.83 | 2.16 | 14.14 | 660333.50† | 48.67 | 72.29 | 64.28 | 163.73 | 116.54 |
| | 7 | 233 | 299.65 | 5.39 | 21.28 | NA | 81.61 | 112.20 | 123.96 | 210.15 | 223.60 |
| | 8 | 232 | 328.84 | 12.14 | 35.06 | NA | 87.69 | 184.51 | 215.95 | 279.48 | 299.35 |
| | 10 | 226 | 514.58 | 66.20 | 130.78 | NA | 163.79 | 295.27 | 492.67 | 394.06 | 650.51 |
| | 12 | 224 | 673.66 | 494.29 | 205.12 | NA | 239.78 | 478.27 | 993.15 | 496.81 | 1112.39 |
| | 15 | 216 | 984.47 | 3960.76 | 120.31 | NA | 467.60 | 884.13 | 2548.69 | 647.87 | 2480.67 |
| | 20 | 220 | 1398.23 | NA | 140.12 | NA | 769.65 | 1990.79 | 7789.37 | 1108.52 | NA |
| 0.01 | 3 | 252 | 0.45 | 0.04 | 1.61 | 0.33 | 0.91 | NA | NA | 0.78 | 1.27 |
| | 4 | 247 | 0.83 | 0.10 | 2.62 | 23.20 | 1.29 | 5.88 | 6.48 | 1.31 | 3.09 |
| | 5 | 243 | 1.45 | 0.24 | 3.03 | 2758.43 | 2.95 | 15.15 | 19.94 | 1.80 | 9.63 |
| | 6 | 238 | 2.54 | 0.36 | 3.85 | 260315.10 | 3.60 | 30.51 | 40.18 | 2.49 | 14.83 |
| | 7 | 233 | 3.04 | 0.69 | 4.57 | NA | 6.47 | 49.27 | 74.55 | 3.07 | 29.02 |
| | 8 | 232 | 3.75 | 1.55 | 5.43 | NA | 6.98 | 76.57 | 125.90 | 4.01 | 44.85 |
| | 10 | 226 | 5.45 | 6.73 | 6.71 | NA | 13.08 | 162.76 | 321.08 | 5.99 | 100.00 |
| | 12 | 224 | 7.32 | 30.86 | 9.48 | NA | 42.43 | 299.07 | 661.58 | 7.81 | 194.10 |
| | 15 | 216 | 9.36 | 145.36 | 11.14 | NA | 41.60 | 616.95 | 1551.00 | 9.86 | 508.27 |
| | 20 | 220 | 14.58 | NA | 17.15 | NA | 81.21 | 1652.68 | 4736.82 | 15.71 | NA |

† Incomplete series

Tables 2 and 3 give numerical summaries of the test runs. Table 2 shows the best method for each of the 252 problems. In cases of ties only one method is recorded. This accounts for the discrepancies with Table 3, which lists for each method the number of times the method had the fastest computation time and the number of times it was within 20% of being the fastest. For further illustration the problems are broken down by the probability $p$.



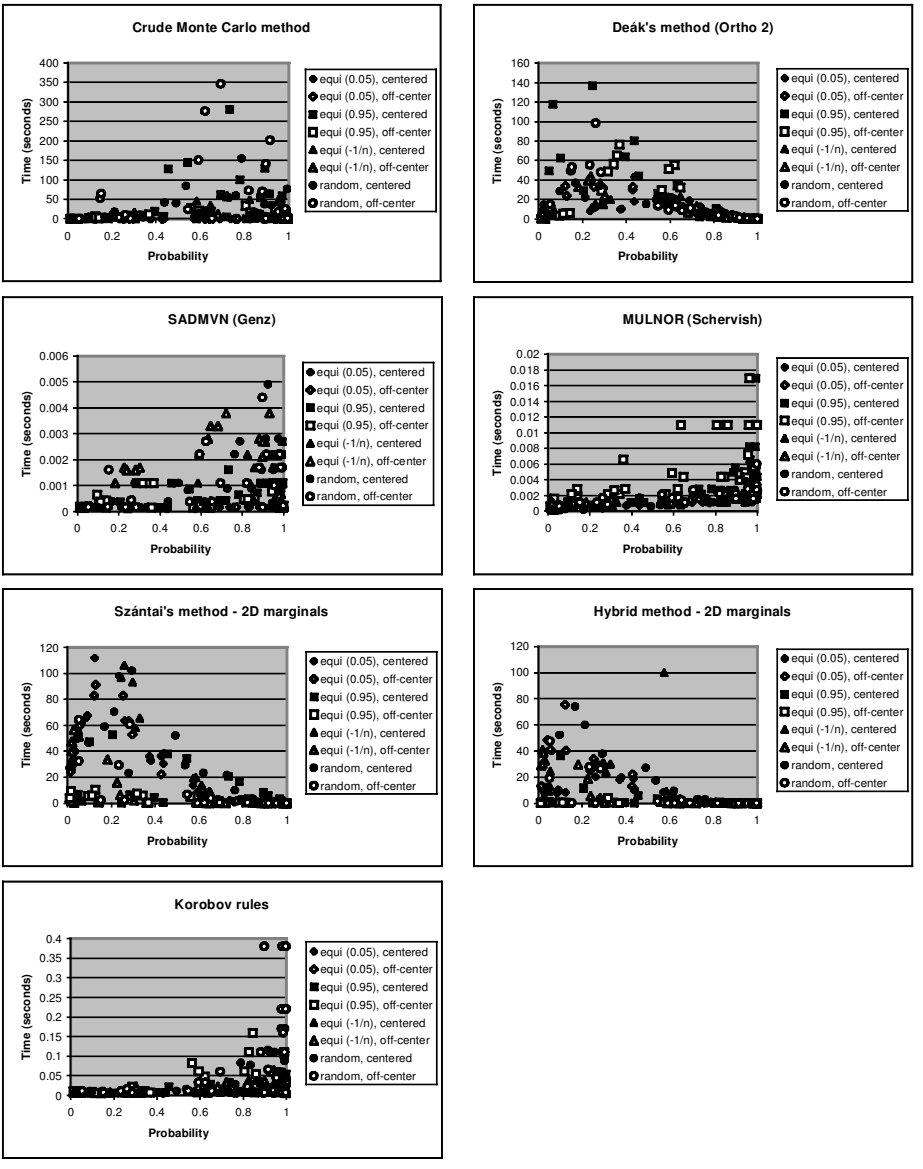


*Figure 2. Computation times for $n = 3$, tolerance $= 0.0001$.*

Table 2. Best Method for $n = 3$

| | | *Accuracy 0.0001 Replication* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| *Matrix* | *Position* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* |
| 0.05 | centered | 2 | 2 | 2 | 2 | 8 | 8 | | |
| | | 2 | 2 | 2 | 2 | 8 | 8 | | |
| | | 2 | 2 | 2 | 2 | 9 | 8 | | |
| | | 2 | 2 | 2 | 2 | 2 | 9 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | 9 | 8 | |
| | off-center | | | 2 | 2 | 2 | 2 | 8 | 9 |
| | | | | 2 | 2 | 2 | 2 | 8 | 8 |
| | | | 2 | 2 | 2 | 2 | 8 | 9 | 8 |
| | | | 2 | 2 | 2 | 2 | 2 | 8 | |
| | | | | 2 | 2 | 2 | 2 | 8 | 8 |
| 0.95 | centered | 2 | 2 | 2 | 2 | 2 | 2 | | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| | off-center | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| | | | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| $-1/n$ | centered | 4 | 2 | 2 | 2 | 2 | 8 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | 8 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | 8 | 8 | |
| | | 2 | 2 | 2 | 2 | 4 | 8 | 8 | |
| | | 2 | 2 | 4 | 4 | 4 | 9 | 9 | |
| | off-center | | 2 | 4 | 4 | 8 | 9 | 9 | |
| | | | | 2 | 2 | 4 | 4 | 8 | 8 |
| | | | | 2 | 4 | 4 | 8 | 8 | 8 |
| | | | | 2 | 4 | 4 | 8 | 8 | 8 |
| | | | | 2 | 4 | 4 | 9 | 8 | |
| random | centered | 2 | 2 | 2 | 2 | 4 | 8 | 9 | |
| | | 4 | 4 | 4 | 4 | 8 | 9 | | |
| | | 2 | 4 | 4 | 4 | 8 | 8 | | |
| | | 2 | 2 | 4 | 2 | 2 | 2 | 2 | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | |
| | off-center | 2 | 2 | 2 | 2 | 2 | 9 | | |
| | | | | | 2 | 8 | 9 | 8 | 8 |
| | | | | | 4 | 4 | 9 | 8 | 8 |
| | | | | 2 | 2 | 2 | 2 | 9 | 8 |
| | | | | | 2 | 2 | 2 | 9 | 2 |

(1 = Crude MC, 2 = SADMVN, 3 = KROMVN, 4 = MULNOR, 5 = Deak—Ortho2, 8 = Szantai—2D, 9 = Hybrid—2D)

Table 3. Comparison of All Methods, $n = 3$

| *Probability* | *<0.3* | | *0.3–0.9* | | *>0.9* | |
|---|---|---|---|---|---|---|
| *Method* | *Best* | *Near best* | *Best* | *Near best* | *Best* | *Near best* |
| Crude MC | 0 | 0 | 0 | 0 | 0 | 0 |
| SADMVN | 57 | 1 | 70 | 2 | 39 | 1 |
| KROMVN | 0 | 0 | 0 | 0 | 0 | 0 |
| MULNOR | 11 | 2 | 17 | 0 | 7 | 0 |
| Deak—Ortho 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| Szantai—2D | 0 | 0 | 2 | 0 | 42 | 7 |
| Hybrid—2D | 0 | 0 | 4 | 0 | 49 | 2 |
| Number of problems | 63 | | 90 | | 99 | |



Both tables clearly show the superiority of Genz's code SADMVN, but they also illustrate the good showing of the bounding methods when $p$ is large. Table 2 also indicates that SADMVN is less successful for asymmetric problems and when the correlation matrix has large negative entries.

Figure 3 records the achieved accuracy for each of the methods. (Since the correlation matrices were all randomly generated, exact values for the probabilities are not available. Instead each method was compared to an average of all the methods.)

None of the methods appear to be without problems. The crude Monte Carlo method does not solve two problems with very large probabilities. The reason may be that no sample points were generated outside the rectangle $(\mathbf{a}, \mathbf{b})$ and thus the variance of the estimator is zero, prematurely triggering a stopping criterion. Genz's code SADMVN occasionally encounters difficulties on problems with at least one large positive correlation. This points to possible numerical problems and was addressed in a follow-up test. Deák's method with crude Monte Carlo sampling occasionally returns no answer, and Szántai's method—and to a lesser extent the hybrid method—fails on problems with very low probabilities. The remaining methods are sometimes inaccurate as well, most notably on one series of problems where the random correlation matrix is extremely ill-conditioned.

A second set of tests was conducted to study the issue of accuracy more carefully. Equicorrelated matrices were generated with correlation coefficients of $0.05, 0.10, \ldots, 0.95$. For these problems highly accurate probabilities can be computed using the method of Tong (1990). The maximum observed errors for some methods are depicted in Figure 4. Schervish's method stays well within the tolerance limit of 0.0001 throughout the range of the experiment, as does Korobov integration. Deák's method is based on sampling, and it, too, performs more or less as expected. The successive approximation procedure, however, shows a marked deterioration of accuracy as the correlation matrix becomes more and more ill-conditioned, giving clear indication of numerical instability.

It thus emerges that SADMVN is to be preferred for problems that are known to be well-behaved and Schervish's method should be used for problems that are known or suspected to be ill-conditioned. If the probability of the region is expected to be large, either Szántai's method or the hybrid method (with two-dimensional marginals) is a reasonable alternative.

## 5.2 THE CASE $n = 4$

The same methodology was used as for the case $n = 3$, resulting in a total of 247 problems. Computation times for all nine methods are graphed in Figure 5. Again only the tightest accuracy requirement (0.0001) is given. Based on the previous experience, Schervish's method was used in the bounding procedures SZ3MVN and HYBRID to compute the three-dimensional marginals. Using the three-dimensional marginals generally

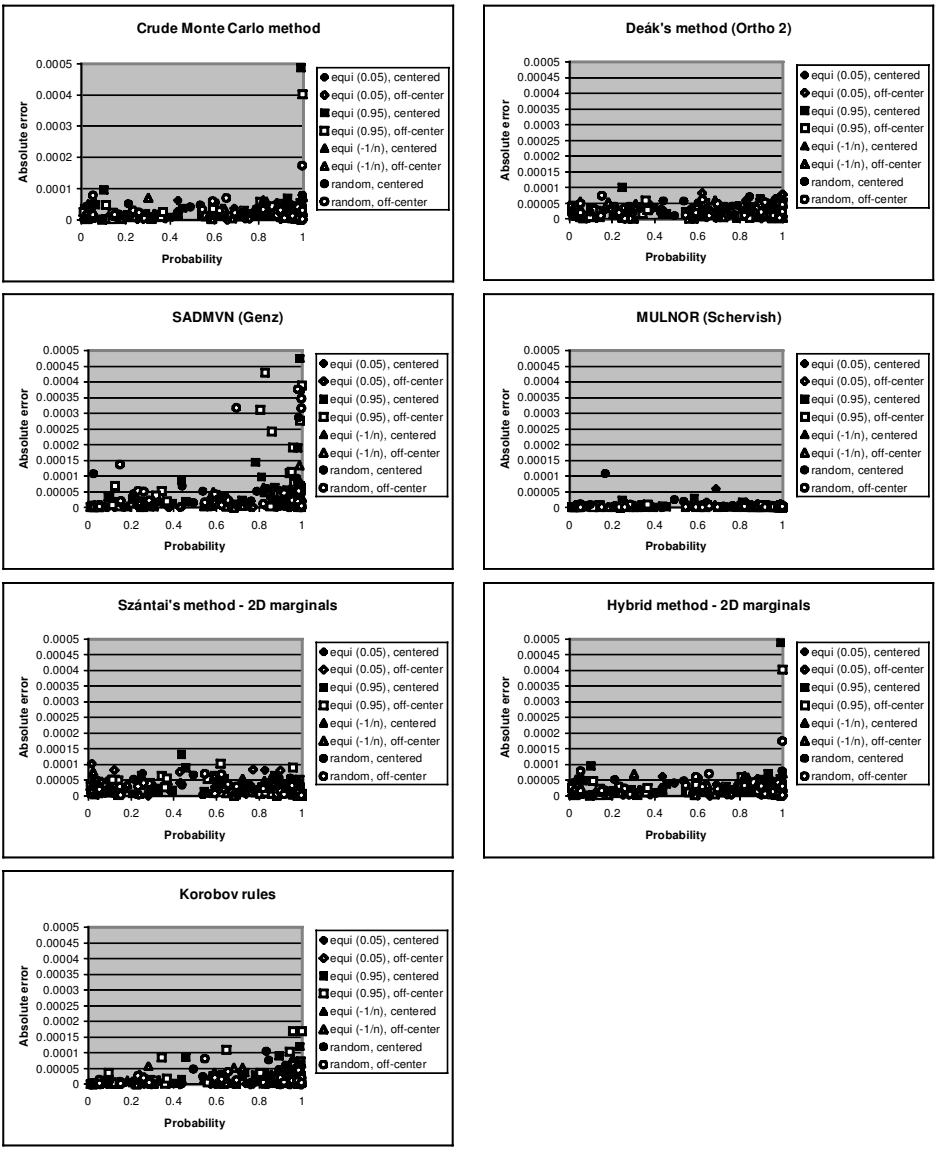


*Figure 3. Estimated achieved accuracy for n = 3.*

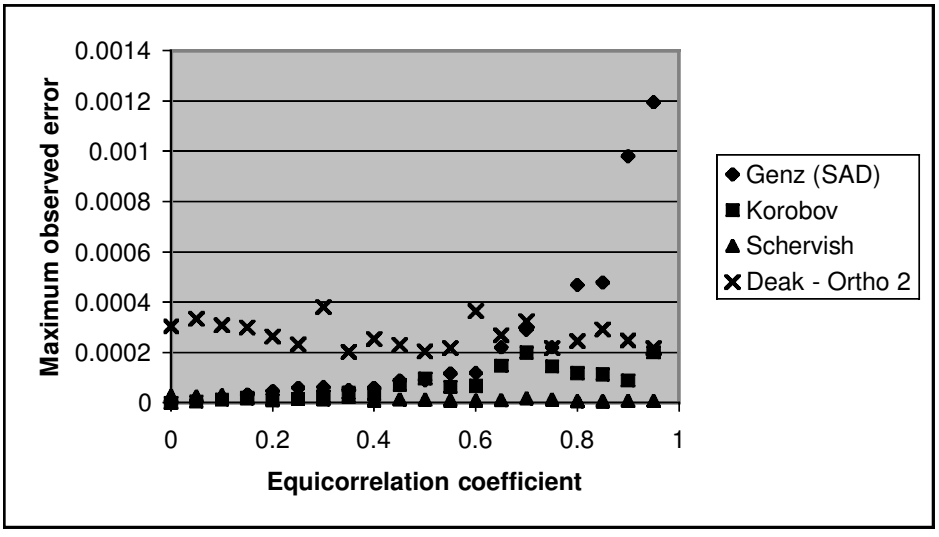


*Figure 4. Maximum observed errors for equicorrelated problems, $n = 3$.*



improved the computation times of both Szántai's method and the hybrid method, at least for small and moderate values of $p$. Additionally, the hybrid methods generally improve on Szántai's approach, although neither is competitive unless the probability $p$ is very close to 1. In this case, even the two-dimensional bounds are so tight that the sampling phase is redundant, and Szántai's method as well as the hybrid method easily post the fastest computation times. On all other problems it is once more apparent that SADMVN is faster than all others (see Tables 4 and 5).

The dependence of computation time on the probability $p$ is the same as for the three-dimensional problems: Computation times for the sampling methods are highest when $p$ is near 0.5, the computation times for the integration method increase with $p$, and the times for the bounding procedures decrease as $p$ increases.

Other observations are that the adaptive partitioning method appears to have trouble with large negative correlations, and Schervish's method and Korobov integration seem to handle centered problems more easily than noncentered ones.

The estimated accuracy (Figure 6) is similar to the case $n = 3$. On problems with very small $p$ the bounding procedures frequently do not return reliable results because no points of interest are generated during the sampling phase and a premature termination criterion is encountered. SADMVN also occasionally returns inaccurate results, which points at numerical difficulties. Figure 7 shows achieved accuracies for equicorrelated problems. (The desired tolerance was set to 0.0001 for these tests.)

As in the case $n = 3$, the performance of SADMVN deteriorates as the equicorrelation coefficient approaches 1, while the other methods tested appear much more robust. For this reason we recommend using one of the bounding procedures when the probability $p$

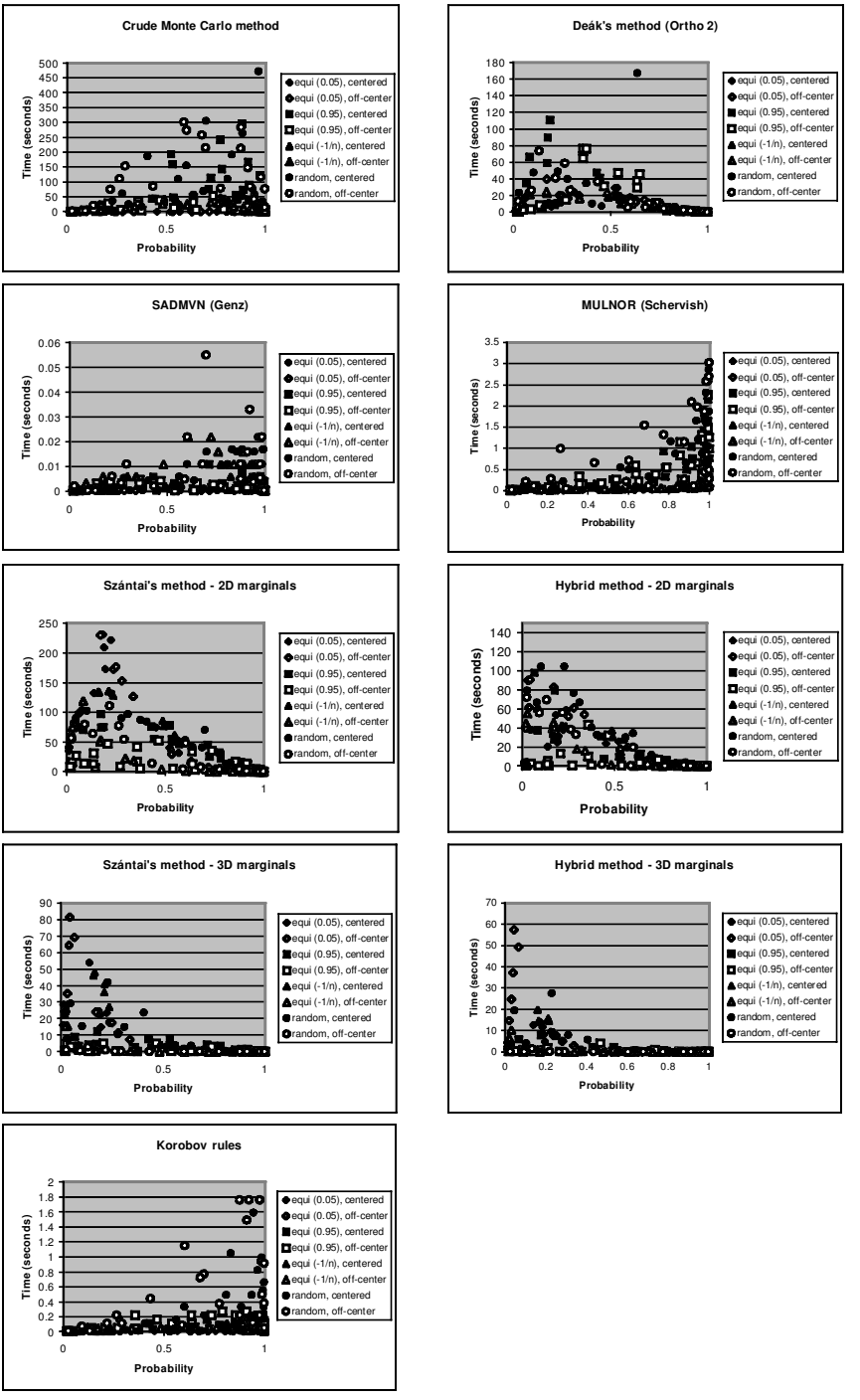


*Figure 5. Computation times for $n = 4$, tolerance $= 0.0001$.*

Table 4. Best Method for $n = 4$

| | | *Accuracy 0.0001 Replication* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| *Matrix* | *Position* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* |
| 0.05 | centered | 2 | 2 | 2 | 2 | 2 | 8 | 8 | |
| | | | 2 | 2 | 2 | 2 | 8 | 9 | |
| | | 2 | 2 | 2 | 2 | 2 | 8 | 9 | |
| | | 2 | 2 | 2 | 2 | 2 | 8 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | 8 | 8 | |
| | off-center | | 2 | 2 | 2 | 2 | 8 | 9 | |
| | | | | 2 | 2 | 2 | 2 | 8 | 9 |
| | | | | 2 | 2 | 2 | 2 | 8 | 8 |
| | | | | 2 | 2 | 2 | 2 | 9 | 9 |
| | | | | 2 | 2 | 2 | 2 | 8 | 8 |
| 0.95 | centered | 2 | 2 | 2 | 2 | 2 | 2 | | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | |
| | off-center | | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| $-1/n$ | centered | 2 | 2 | 2 | 2 | 2 | 9 | 9 | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 8 | |
| | | | 2 | 2 | 2 | 2 | 2 | 9 | |
| | | 2 | 2 | 2 | 2 | 2 | 9 | 8 | |
| | | | 2 | 2 | 2 | 2 | 2 | 8 | |
| | off-center | | | 2 | 2 | 6 | 8 | 8 | |
| | | | | | 2 | 6 | 2 | 8 | 8 |
| | | | | | 2 | 6 | 2 | 8 | 8 |
| | | | | 2 | 2 | 7 | 8 | 9 | 8 |
| | | | | | 2 | 2 | 2 | 8 | 8 |
| random | centered | 2 | 2 | 2 | 7 | 6 | 9 | | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | |
| | | 2 | 2 | 2 | 2 | 2 | 8 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | | | |
| | off-center | | | 2 | 2 | 2 | 2 | 9 | 9 |
| | | | | 2 | 2 | 6 | 7 | 8 | 8 |
| | | | | | 2 | 2 | 2 | 8 | 9 |
| | | | | | 2 | 2 | 2 | 9 | 8 |
| | | | | | 2 | 6 | 2 | 2 | 8 |

(1 = Crude MC, 2 = SADMVN, 3 = KROMVN, 4 = MULNOR, 5 = Deak—Ortho2, 6 = Szantai 7 = Hybrid, 8 = Szantai—2D, 9 = Hybrid—2D)

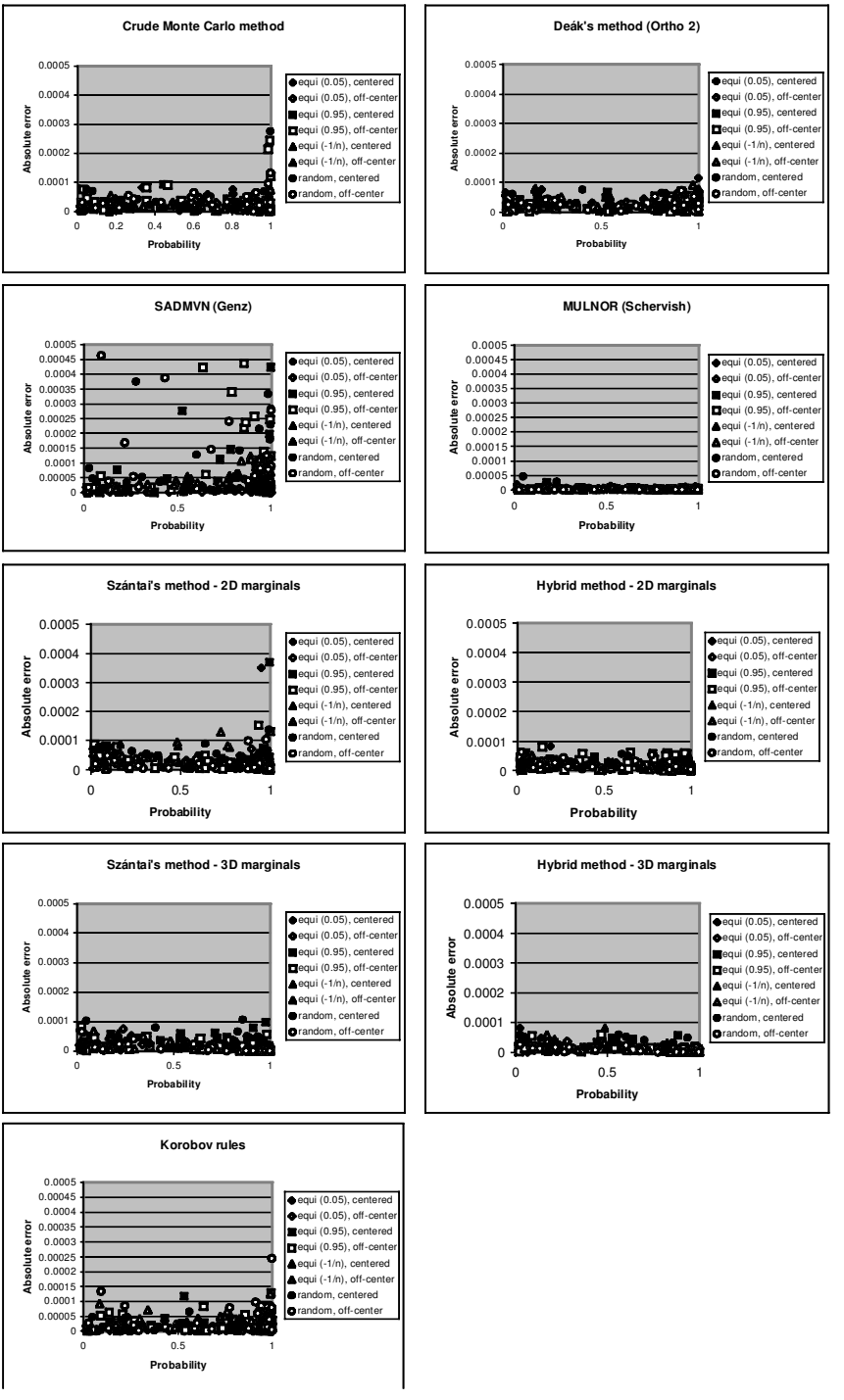


*Figure 6. Estimated achieved accuracy for $n = 4$.*

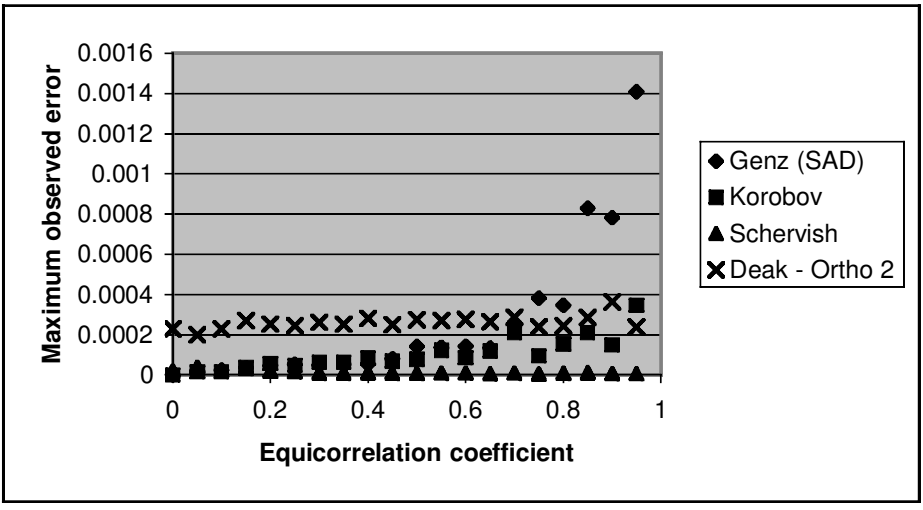


*Figure 7. Maximum observed errors for equicorrelated problems, $n = 4$.*

is close to 1 and to use SADMVN otherwise, provided that the correlation matrix is not too ill-conditioned. Should the accuracy of SADMVN ever be in doubt, a check can be made using Schervish's method. Figure 7 suggests that this method is much more accurate than the nominal error tolerance would indicate.

### 5.3 FIVE- TO EIGHT-DIMENSIONAL PROBLEMS

These problems are grouped together because, by and large, the methods behave similarly. It is difficult to display the computation times because there are so many problems,

Table 5. Comparison of All Methods, $n = 4$

| *Probability* | *< 0.3* | | *0.3–0.9* | | *> 0.9* | |
|---|---|---|---|---|---|---|
| *Method* | *Best* | *Near best* | *Best* | *Near best* | *Best* | *Near best* |
| Crude MC | 0 | 0 | 0 | 0 | 0 | 0 |
| SADMVN | 59 | 0 | 77 | 0 | 53 | 1 |
| KROMVN | 0 | 0 | 0 | 0 | 0 | 0 |
| MULNOR | 0 | 0 | 0 | 0 | 0 | 0 |
| Deak—Ortho 2 | 0 | 0 | 0 | 0 | 0 | 1 |
| Szantai | 0 | 0 | 8 | 0 | 2 | 0 |
| Hybrid | 0 | 0 | 8 | 1 | 2 | 0 |
| Szantai—2D | 0 | 0 | 0 | 0 | 35 | 7 |
| Hybrid—2D | 0 | 0 | 0 | 0 | 33 | 11 |
| Number of problems | 59 | | 84 | | 104 | |

Table 6. Cumulative Performance by Method, $n$ = 5, 6, 7, 8

| | | | | crude MC | | SADMVN | | KROMVN | | MULNOR | | Deak - ortho 2 | | Szantai | | Hybrid | | Szantai - 2D | | Hybrid - 2D | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | N of | | near | | near | | near | | near | | near | | near | | near | | near | | near |
| $n$ | Accuracy | Probability | probs | is best | best | is best | best | is best | best | is best | best | is best | best | is best | best | is best | best | is best | best | is best | best |
| | | <0.3 | 55 | 2 | 1 | 55 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0.0001 | 0.3-0.9 | 85 | 0 | 0 | 78 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 5 | 1 | 10 | 3 | 0 | 0 | 0 | 0 |
| | | >0.9 | 103 | 0 | 0 | 55 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 5 | 2 | 31 | 6 | 40 | 4 |
| | | <0.3 | 55 | 4 | 5 | 54 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.001 | 0.3-0.9 | 85 | 8 | 6 | 76 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 2 | 2 |
| | | >0.9 | 103 | 0 | 1 | 37 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 49 | 8 | 56 | 10 |
| | | <0.3 | 55 | 25 | 7 | 44 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.01 | 0.3-0.9 | 85 | 4 | 2 | 63 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 0 | 14 | 5 |
| | | >0.9 | 103 | 7 | 1 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 59 | 15 | 61 | 15 |
| | | <0.3 | 51 | 3 | 0 | 49 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0.0001 | 0.3-0.9 | 89 | 0 | 0 | 75 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 12 | 3 | 0 | 0 | 0 | 0 |
| | | >0.9 | 98 | 1 | 0 | 49 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 2 | 9 | 2 | 25 | 13 | 27 | 13 |
| | | <0.3 | 51 | 11 | 0 | 41 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.001 | 0.3-0.9 | 89 | 10 | 1 | 77 | 1 | 5 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 2 | 1 | 0 | 0 |
| | | >0.9 | 98 | 6 | 0 | 35 | 1 | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 1 | 0 | 0 | 29 | 29 | 48 | 10 |
| | | <0.3 | 51 | 33 | 2 | 28 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.01 | 0.3-0.9 | 89 | 17 | 2 | 64 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 6 | 13 | 0 |
| | | >0.9 | 98 | 7 | 4 | 21 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 47 | 30 | 70 | 8 |
| | | <0.3 | 47 | 4 | 0 | 40 | 2 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0.0001 | 0.3-0.9 | 93 | 0 | 0 | 70 | 4 | 6 | 1 | 0 | 0 | 0 | 0 | 8 | 1 | 15 | 2 | 1 | 0 | 1 | 0 |
| | | >0.9 | 93 | 0 | 0 | 46 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 9 | 4 | 6 | 5 | 23 | 13 | 33 | 2 |
| | | <0.3 | 47 | 12 | 0 | 33 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.001 | 0.3-0.9 | 93 | 15 | 1 | 68 | 2 | 7 | 1 | 0 | 0 | 1 | 0 | 7 | 0 | 1 | 1 | 5 | 1 | 0 | 0 |
| | | >0.9 | 93 | 1 | 0 | 33 | 1 | 0 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 0 | 0 | 46 | 11 | 48 | 8 |
| | | <0.3 | 47 | 36 | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.01 | 0.3-0.9 | 93 | 15 | 0 | 59 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 5 | 17 | 1 |
| | | >0.9 | 93 | 6 | 2 | 15 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 49 | 17 | 62 | 9 |
| | | <0.3 | 53 | 4 | 0 | 45 | 0 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0.0001 | 0.3-0.9 | 85 | 0 | 0 | 56 | 2 | 15 | 2 | 0 | 0 | 0 | 0 | 11 | 0 | 7 | 2 | 1 | 0 | 0 | 0 |
| | | >0.9 | 94 | 0 | 0 | 49 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 9 | 0 | 7 | 1 | 26 | 2 | 19 | 7 |
| | | <0.3 | 53 | 18 | 0 | 27 | 2 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.001 | 0.3-0.9 | 85 | 13 | 1 | 53 | 3 | 11 | 2 | 0 | 0 | 5 | 1 | 3 | 2 | 1 | 1 | 4 | 0 | 0 | 0 |
| | | >0.9 | 94 | 1 | 0 | 27 | 2 | 0 | 3 | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 46 | 1 | 26 | 28 |
| | | <0.3 | 53 | 43 | 2 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.01 | 0.3-0.9 | 85 | 18 | 1 | 47 | 1 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 18 | 2 | 4 | 3 |
| | | >0.9 | 94 | 13 | 1 | 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 62 | 4 | 40 | 19 |

Table 7. Best method for $n = 8$

| | | Accuracy 0.0001 Replication | | | | | | | Accuracy 0.001 Replication | | | | | | | Accuracy 0.01 Replication | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Matrix* | *Position* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *1* | *2* | *3* | *4* | *5* | *6* | *7* |
| 0.05 | centered | 1 | 2 | 2 | 3 | 2 | 8 | | 1 | 1 | 2 | 1 | 8 | 8 | | 1 | 1 | 1 | 8 | 9 | 9 | |
| | | 1 | 2 | 2 | 3 | 2 | 8 | | 1 | 1 | 1 | 1 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | | 2 | 2 | 3 | 3 | 2 | 8 | | 1 | 2 | 2 | 1 | 8 | 9 | | 1 | 1 | 1 | 8 | 9 | 8 | |
| | | 1 | 2 | 3 | 3 | 2 | 8 | | 1 | 1 | 2 | 1 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | | 1 | 2 | 3 | 3 | 2 | 8 | | 1 | 1 | 2 | 1 | 8 | 9 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | off-center | 2 | 2 | 2 | 2 | 8 | | | 1 | 1 | 1 | 8 | 8 | | | 1 | 1 | 1 | 8 | 8 | | |
| | | 2 | 3 | 3 | 2 | 9 | | | 1 | 1 | 1 | 9 | 8 | | | 1 | 1 | 1 | 8 | 8 | | |
| | | 2 | 3 | 3 | 2 | 8 | | | 1 | 2 | 1 | 8 | 8 | | | 1 | 1 | 8 | 8 | 8 | | |
| | | 2 | 2 | 2 | 2 | 8 | | | 1 | 1 | 1 | 8 | 8 | | | 1 | 1 | 8 | 9 | 8 | | |
| | | 2 | 3 | 3 | 2 | 8 | | | 1 | 2 | 1 | 8 | 8 | | | 1 | 1 | 8 | 8 | 9 | | |
| 0.95 | centered | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 5 | 2 | 1 | 2 | 2 | 2 | 2 | 8 | 9 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | 2 | 2 | 2 | 2 | 6 | 2 | | 2 | 2 | 2 | 2 | 2 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | 2 | 2 | 2 | 2 | 6 | 2 | | 1 | 2 | 2 | 2 | 2 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 6 | 2 | 1 | 2 | 2 | 2 | 2 | 8 | 9 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 9 | 2 | 1 | 2 | 2 | 2 | 2 | 1 | 1 |
| | off-center | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 2 | 5 | 8 | 1 | 2 | 2 | 2 | 2 | 1 | 1 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | 2 | 3 | 2 | 2 | 2 | 2 | | 1 | 1 | 1 | 1 | 1 | 1 | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 5 | 2 | 1 | 2 | 2 | 2 | 2 | 8 | 1 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 2 | 2 | 8 | 1 | 2 | 2 | 2 | 1 | 1 | 1 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 2 | 2 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | |
| $-1/n$ | centered | 2 | 2 | 2 | 2 | 8 | 8 | | 1 | 2 | 3 | 8 | 9 | 9 | | 1 | 2 | 2 | 2 | 8 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | 9 | | 2 | 2 | 2 | 2 | 9 | 8 | | 1 | 2 | 2 | 9 | 8 | 8 | |
| | | 2 | 2 | 3 | 6 | 2 | 8 | | 2 | 2 | 2 | 2 | 9 | 8 | | 1 | 2 | 2 | 8 | 8 | 8 | |
| | | 2 | 2 | 6 | 7 | 2 | 8 | | 2 | 2 | 3 | 2 | 9 | 9 | | 1 | 2 | 2 | 8 | 8 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | 8 | | 2 | 2 | 2 | 2 | 9 | 9 | | 1 | 2 | 2 | 8 | 8 | 8 | |
| | off-center | | 3 | 6 | 8 | 8 | | | | 3 | 3 | 8 | 8 | | | | 1 | 2 | 8 | 8 | | |
| | | 3 | 3 | 6 | 2 | 8 | | | 2 | 3 | 3 | 8 | 8 | | | 1 | 1 | 9 | 8 | 8 | | |
| | | 3 | 7 | 6 | 8 | 9 | | | 3 | 3 | 2 | 8 | 8 | | | 1 | 2 | 8 | 9 | 8 | | |
| | | 2 | 6 | 6 | 9 | 9 | | | 3 | 3 | 8 | 8 | 8 | | | 1 | 2 | 8 | 9 | 8 | | |
| | | 2 | 3 | 8 | 8 | 8 | | | 2 | 3 | 8 | 9 | 8 | | | 1 | 2 | 8 | 8 | 8 | | |
| random | centered | 2 | 2 | 2 | 7 | 6 | 2 | | 3 | 2 | 6 | 2 | 2 | 8 | | 2 | 3 | 8 | 2 | 8 | 8 | |
| | | 2 | 2 | 6 | 6 | 6 | 6 | | 2 | 2 | 5 | 6 | 2 | 8 | | 2 | 2 | 2 | 2 | 8 | 8 | |
| | | 2 | 2 | 2 | 2 | 2 | 2 | | 2 | 2 | 2 | 2 | 2 | 8 | | 1 | 2 | 2 | 2 | 8 | 9 | |
| | | 2 | 2 | 2 | 2 | 7 | 2 | | 2 | 2 | 2 | 2 | 6 | 8 | | 2 | 2 | 2 | 2 | 8 | 8 | |
| | | 2 | 2 | 2 | 7 | 6 | 5 | | 2 | 2 | 2 | 2 | 2 | 8 | | 2 | 2 | 2 | 2 | 8 | 8 | |
| | off-center | 2 | 2 | 2 | 9 | 8 | | | 2 | 2 | 5 | 8 | 8 | | | 2 | 2 | 8 | 8 | 8 | | |
| | | 2 | 2 | 7 | 6 | 8 | | | 2 | 2 | 5 | 8 | 9 | | | 1 | 3 | 8 | 8 | 8 | | |
| | | 2 | 2 | 6 | 6 | 8 | | | 2 | 2 | 5 | 8 | 9 | | | 1 | 2 | 2 | 8 | 8 | | |
| | | 2 | 2 | 7 | 6 | 8 | | | 3 | 5 | 5 | 8 | 8 | | | 2 | 9 | 2 | 8 | 8 | | |
| | 2 | 2 | 2 | 6 | 8 | | | | 2 | 2 | 5 | 2 | 8 | | | 2 | 2 | 8 | 8 | 8 | | |

(1 = Crude MC, 2 = SADMVN, 3 = KROMVN, 4 = MULNOR, 5 = Deak—Ortho2, 6 = Szantai, 7 = Hybrid, 8 = Szantai—2D, 9 = Hybrid—2D)

and the times differ so much from one problem to another that graphical displays would be nearly impossible to read. For that reason we only quote summary results. In Table 6 we count how many times a method is best or nearly so. Again more than 200 problems were solved for each $n$, and since the computation times have increased sufficiently it is possible to report the results for three different accuracy requirements, 0.01, 0.001, and 0.0001.

Genz's code SADMVN still solves a good many of the problems in the fastest time, but for small values of $p$ the crude Monte Carlo method must be considered as a serious alternative, especially if the accuracy requirement is not very high. At the other end of the spectrum, if $p$ is large, the bounding methods with two-dimensional marginals perform very well. For mid-sized problems Deák's method is occasionally fastest.

At the next level of detail, we study the effect of the correlation matrix. As stated before, the results differ little with $n$, and Table 7, which shows the breakdown for $n = 8$, will have to suffice as an illustration. SADMVN performs fastest on the problems with large positive correlations. (Those are unfortunately also the problems on which its accuracy is doubtful.) The bounding methods perform best on the completely random problems and problems with negative correlations when $p$ is large; on problems with highly positively correlated components they are occasionally slow and inaccurate. Nearly independent problems and also occasionally negatively correlated problems with moderate $p$ are best handled by Korobov integration if high accuracy is desired. Deák's method, finally, is best on the completely random problems when both $p$ and the accuracy requirement are moderate. (It is interesting to note that those are also the problems on which Deák's method is expected to take longer than on problems with small or large probabilities. Evidently the slowdown is not as bad as for most of the other methods.)

### 5.4 10- TO 20-DIMENSIONAL PROBLEMS

In order to test the limits of computability, we performed the same series of tests for selected problems in higher dimensions. We chose $n = 10$, 12, 15, and 20 and report the findings similarly in two tables. Table 8 is the equivalent of Table 6 and gives the cumulative performance for each method. It clearly shows the limits of the adaptive partitioning method, which are reached at $n = 12$. For larger $n$, the crude Monte Carlo method looks better and better, particularly if the probability $p$ is small. For large $p$, the bounding methods with two-dimensional marginals are the methods of choice, while Korobov integration looks most promising for midrange problems. Occasionally Deák's method is best.

In Table 9 the results are again broken down problem by problem. We again chose to show the results for only one size, $n = 20$. Table 9 indicates that the success of Deák's method is mostly limited to the completely random problems, while Korobov integration works best on nearly independent correlation matrices and those with large positive coefficients.

Table 8. Cumulative Performance by Method, $n = 10, 12, 15, 20$

| | | | | crude MC | | SADMVN | | KROMVN | | MULNOR | | Deak - ortho 2 | | Szantai | | Hybrid | | Szantai - 2D | | Hybrid - 2D | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | N of | | near | | near | | near | | near | | near | | near | | near | | near | | near |
| *n* | Accuracy | Probability | probs | is best | best | is best | best | is best | best | is best | best | is best | best | is best | best | is best | best | is best | best | is best | best |
| | | <0.3 | 51 | 3 | 0 | 34 | 1 | 13 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 |
| 10 | 0.0001 | 0.3-0.9 | 85 | 0 | 0 | 43 | 1 | 20 | 1 | 0 | 0 | 0 | 0 | 12 | 0 | 10 | 1 | 0 | 0 | 0 | 0 |
| | | >0.9 | 90 | 1 | 0 | 32 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 10 | 4 | 5 | 4 | 36 | 1 | 17 | 5 |
| | | <0.3 | 51 | 19 | 0 | 16 | 1 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.001 | 0.3-0.9 | 85 | 20 | 0 | 26 | 1 | 25 | 5 | 0 | 0 | 6 | 3 | 10 | 0 | 1 | 1 | 4 | 3 | 0 | 0 |
| | | >0.9 | 90 | 1 | 0 | 15 | 2 | 6 | 0 | 0 | 0 | 7 | 4 | 0 | 0 | 0 | 0 | 54 | 5 | 37 | 1 |
| | | <0.3 | 51 | 51 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.01 | 0.3-0.9 | 85 | 30 | 0 | 34 | 0 | 9 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | 1 | 7 | 0 |
| | | >0.9 | 90 | 15 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52 | 14 | 47 | 3 |
| | | <0.3 | 54 | 5 | 0 | 2 | 2 | 42 | 2 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0.0001 | 0.3-0.9 | 75 | 0 | 0 | 17 | 0 | 32 | 2 | 0 | 0 | 1 | 2 | 18 | 3 | 8 | 3 | 0 | 0 | 0 | 0 |
| | | >0.9 | 95 | 0 | 0 | 26 | 0 | 14 | 0 | 0 | 0 | 0 | 0 | 10 | 2 | 1 | 1 | 35 | 2 | 19 | 3 |
| | | <0.3 | 54 | 19 | 1 | 0 | 0 | 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.001 | 0.3-0.9 | 75 | 15 | 0 | 6 | 2 | 36 | 3 | 0 | 0 | 8 | 2 | 8 | 2 | 1 | 1 | 2 | 2 | 0 | 0 |
| | | >0.9 | 95 | 6 | 0 | 1 | 0 | 14 | 0 | 0 | 0 | 7 | 2 | 0 | 0 | 0 | 0 | 56 | 7 | 25 | 5 |
| | | <0.3 | 54 | 52 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.01 | 0.3-0.9 | 75 | 31 | 4 | 0 | 1 | 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 5 | 2 | 1 |
| | | >0.9 | 95 | 20 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 67 | 8 | 40 | 9 |
| | | <0.3 | 51 | 3 | 0 | 0 | 0 | 25 | 3 | 0 | 0 | 0 | 0 | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 0.0001 | 0.3-0.9 | 69 | 0 | 0 | 4 | 0 | 20 | 2 | 0 | 0 | 0 | 0 | 45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | >0.9 | 96 | 0 | 0 | 15 | 0 | 11 | 2 | 0 | 0 | 0 | 0 | 37 | 7 | 0 | 0 | 30 | 8 | 12 | 10 |
| | | <0.3 | 51 | 17 | 0 | 0 | 0 | 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.001 | 0.3-0.9 | 69 | 16 | 0 | 2 | 0 | 42 | 0 | 0 | 0 | 1 | 3 | 5 | 1 | 0 | 0 | 4 | 2 | 0 | 0 |
| | | >0.9 | 96 | 6 | 0 | 10 | 0 | 17 | 1 | 0 | 0 | 3 | 1 | 0 | 0 | 0 | 0 | 55 | 4 | 18 | 19 |
| | | <0.3 | 51 | 52 | 1 | 0 | 0 | 8 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| | 0.01 | 0.3-0.9 | 69 | 34 | 3 | 7 | 0 | 16 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 2 | 0 | 0 |
| | | >0.9 | 96 | 18 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 66 | 13 | 37 | 22 |
| | | <0.3 | 53 | 5 | 2 | 0 | 0 | 39 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0.0001 | 0.3-0.9 | 85 | 0 | 0 | 0 | 2 | 53 | 0 | 0 | 0 | 13 | 0 | 0 | 0 | 0 | 0 | 4 | 1 | 0 | 0 |
| | | >0.9 | 94 | 2 | 0 | 9 | 2 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 64 | 0 | 0 | 0 |
| | | <0.3 | 53 | 23 | 2 | 0 | 0 | 32 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.001 | 0.3-0.9 | 85 | 15 | 0 | 0 | 0 | 38 | 3 | 0 | 0 | 6 | 2 | 2 | 3 | 0 | 2 | 7 | 2 | 0 | 0 |
| | | >0.9 | 94 | 9 | 0 | 0 | 0 | 18 | 2 | 0 | 0 | 10 | 1 | 0 | 0 | 0 | 0 | 75 | 1 | 0 | 0 |
| | | <0.3 | 53 | 48 | 1 | 0 | 0 | 9 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.01 | 0.3-0.9 | 85 | 34 | 2 | 0 | 0 | 23 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 2 | 0 | 0 |
| | | >0.9 | 94 | 20 | 2 | 0 | 0 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 85 | 2 | 0 | 0 |

Table 9. Best Method for $n = 20$

| | | *Accuracy 0.0001* | | | | | | | *Accuracy 0.001* | | | | | | | *Accuracy 0.01* | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *Replication* | | | | | | | *Replication* | | | | | | | *Replication* | | | | | | |
| *matrix* | *position* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *1* | *2* | *3* | *4* | *5* | *6* | *7* |
| 0.05 | centered | 1 | 3 | 3 | 3 | 8 | 8 | | 1 | 1 | 1 | 1 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | | 1 | 3 | 3 | 3 | 1 | 8 | | 1 | 1 | 1 | 1 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | | 1 | 3 | 3 | 3 | 8 | 8 | | 1 | 1 | 1 | 1 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | | 1 | 3 | 3 | 3 | 8 | 8 | | 1 | 1 | 1 | 1 | 1 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | | 1 | 3 | 3 | 3 | 8 | 8 | | 1 | 1 | 1 | 1 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | off-center | | 3 | 3 | 3 | 8 | | | | 1 | 1 | 1 | 8 | | | | 1 | 1 | 8 | 8 | | |
| | | | 3 | 3 | 3 | 8 | | | | 1 | 1 | 1 | 8 | | | | 1 | 1 | 8 | 8 | | |
| | | 3 | 3 | 3 | 3 | 8 | | | 1 | 1 | 1 | 8 | 8 | | | 1 | 1 | 8 | 8 | 8 | | |
| | | | 3 | 3 | 3 | 8 | | | | 1 | 1 | 1 | 8 | | | | 1 | 1 | 8 | 8 | | |
| | | | 3 | 3 | 3 | 8 | | | | 1 | 1 | 1 | 8 | | | | 1 | 1 | 8 | 8 | | |
| 0.95 | centered | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 1 | 3 | 3 | 3 | 3 | 3 | 8 | 1 | 3 | 3 | 3 | 8 | 1 | 8 |
| | | 3 | 3 | 3 | 3 | 3 | 8 | | 3 | 3 | 3 | 3 | 5 | 8 | | 3 | 3 | 3 | 8 | 8 | 1 | |
| | | 3 | 3 | 3 | 3 | 3 | 2 | | 3 | 3 | 3 | 3 | 3 | 8 | | 3 | 3 | 3 | 8 | 1 | 1 | |
| | | 3 | 3 | 3 | 3 | 3 | 2 | | 3 | 3 | 3 | 5 | 3 | 8 | | 3 | 3 | 3 | 3 | 8 | 8 | |
| | | 3 | 3 | 3 | 3 | 3 | 2 | | 3 | 3 | 3 | 3 | 5 | 3 | | 3 | 3 | 3 | 8 | 1 | 1 | |
| | off-center | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 1 | 1 | 3 | 1 | 1 | 1 | 1 |
| | | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 5 | 3 | 8 | 1 | 1 | 3 | 3 | 1 | 1 | 1 |
| | | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 8 | 1 | 1 | 1 | 3 | 1 | 1 | 1 |
| | | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 3 | 5 | 8 | 1 | 1 | 1 | 3 | 3 | 1 | 1 |
| $-1/n$ | centered | 3 | 3 | 3 | 8 | 8 | 8 | | 1 | 3 | 8 | 8 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | | 3 | 3 | 3 | 8 | 8 | 8 | | 3 | 3 | 3 | 8 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | | 3 | 3 | 3 | 8 | 8 | 8 | | 3 | 3 | 3 | 8 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | | 3 | 3 | 3 | 8 | 8 | 8 | | 1 | 3 | 3 | 8 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | | 3 | 3 | 3 | 8 | 8 | 8 | | 3 | 3 | 3 | 8 | 8 | 8 | | 1 | 1 | 1 | 8 | 8 | 8 | |
| | off-center | | 3 | 3 | 8 | 8 | | | | 3 | 3 | 8 | 8 | | | | 1 | 1 | 8 | 8 | | |
| | | 3 | 3 | 3 | 8 | 8 | | | 3 | 3 | 3 | 8 | 8 | | | 1 | 1 | 1 | 8 | 8 | | |
| | | | 3 | 3 | 8 | 8 | | | | 3 | 3 | 8 | 8 | | | | 1 | 1 | 8 | 8 | | |
| | | 3 | 3 | 3 | 8 | 8 | | | 1 | 3 | 3 | 8 | 8 | | | 1 | 1 | 1 | 8 | 8 | | |
| | | 3 | 3 | 3 | 8 | 8 | | | 1 | 3 | 3 | 8 | 8 | | | 1 | 1 | 1 | 8 | 8 | | |
| random | centered | 3 | 5 | 5 | 8 | 8 | 8 | | 3 | 3 | 6 | 8 | 8 | 8 | | 1 | 3 | 8 | 8 | 8 | 8 | |
| | | 3 | 5 | 5 | 8 | 8 | 8 | | 3 | 3 | 6 | 8 | 8 | 8 | | 1 | 3 | 8 | 8 | 8 | 8 | |
| | | 3 | 3 | 5 | 8 | 8 | 8 | | 3 | 3 | 8 | 8 | 8 | 8 | | 1 | 8 | 8 | 8 | 8 | 8 | |
| | | 3 | 5 | 5 | 8 | 8 | 8 | | 3 | 5 | 8 | 8 | 8 | 8 | | 1 | 3 | 8 | 8 | 8 | 8 | |
| | | 3 | 5 | 5 | 8 | 8 | 8 | | 3 | 3 | 8 | 8 | 8 | 8 | | 1 | 3 | 8 | 8 | 8 | 8 | |
| | off-center | 5 | 5 | 8 | 8 | | | | 3 | 3 | 8 | 8 | | | | 1 | 8 | 8 | 8 | | | |
| | | 5 | 5 | 8 | 8 | | | | 3 | 5 | 8 | 8 | | | | 3 | 3 | 8 | 8 | | | |
| | | 5 | 5 | 8 | 8 | | | | 3 | 5 | 8 | 8 | | | | 3 | 3 | 8 | 8 | | | |
| | | 3 | 3 | 8 | 8 | | | | 3 | 5 | 8 | 8 | | | | 3 | 8 | 8 | 8 | | | |
| | | 3 | 5 | 8 | 8 | | | | 3 | 5 | 8 | 8 | | | | 3 | 3 | 8 | 8 | | | |

(1 = Crude MC, 2 = SADMVN, 3 = KROMVN, 4 = MULNOR, 5 = Deak—Ortho2, 6 = Szantai, 7 = Hybrid, 8 = Szantai—2D, 9 = Hybrid—2D)

Table 10. Optimal Computation Times

| *Accuracy* | *n* | *5* | *6* | *7* | *8* | *10* | *12* | *15* | *20* |
|---|---|---|---|---|---|---|---|---|---|
| 0.0001 | Cumulative | 4.45 | 16.11 | 48.00 | 114.09 | 524.56 | 1551.68 | 1078.18 | 6359.06 |
| | Maximum | 0.60 | 1.93 | 4.62 | 30.21 | 56.14 | 143.19 | 104.72 | 354.87 |
| 0.001 | Cumulative | 0.57 | 1.88 | 4.17 | 7.88 | 21.08 | 35.11 | 49.23 | 114.17 |
| | Maximum | 0.044 | 0.16 | 0.33 | 0.60 | 1.26 | 2.14 | 4.28 | 7.30 |
| 0.01 | Cumulative | 0.17 | 0.28 | 0.46 | 0.85 | 1.68 | 2.73 | 3.96 | 6.30 |
| | Maximum | 0.003 | 0.008 | 0.011 | 0.028 | 0.044 | 0.055 | 0.11 | 0.16 |



## 6. CONCLUSIONS

We have shown that the best method for finding multivariate normal rectangular probabilities depends on several problem characteristics such as the size of the rectangle and the structure of the matrix. Some preprocessing is therefore necessary to help identify the correct method to use. This does not have to be very sophisticated. For instance, the Cholesky factor of the correlation matrix can be found in $O(n^3)$ operations and is needed by every one of the methods. In the process it is easy to obtain the condition number of the correlation matrix, which may signal impending numerical difficulties and narrows the choice of available methods. In addition the problem dimensions are known, and it is easy to check whether the origin is contained in the rectangle or not. We finally advocate generating a few trial points, no more than 10 or 20, to get a preliminary indication of the size of the probability $p$. These observations guide the selection of the final method to use and should cut down on the computation time in many cases.

But how good is the best method? Table 10 tries to give a short answer to this question. For each of the 250 or so problems in our test battery we selected the method that gave the fastest computation time and report two statistics. We list the cumulative time to solve all the problems, along with the longest time observed in all runs. As can be seen, the numbers are quite impressive: It is possible to obtain four-digit accuracy for a 20-dimensional problem in less than six minutes on a fairly ordinary desktop computer.

## ACKNOWLEDGMENTS

This research was supported in part through a grant from the National Science and Engineering Research Council of Canada. The first author is supported in part by a grant from the National Science and Engineering Research Council of Canada (NSERC). A. Genz and M. Schervish were kind enough to make their software available for inclusion in the numerical tests. The authors also thank the associate editor, A. Buja, and two anonymous referees for their careful reading of an earlier draft. Their thoughtful remarks have served to greatly improve this article.

*[Received May 2000. Revised June 2001.]*

## REFERENCES


Anderson, J. A., and Pemberton, J. D. (1985), "The Grouped Continuous Model for Multivariate Ordered Categorical Variables and Covariate Adjustment," *Biometrics*, 41, 875–885.

Beckers, M., and Haegemans, A. (1992), "Comparison of Numerical Integration Techniques for Multivariate Normal Integrals," Working Paper, Department of Computer Science, Catholic University of Leuven, Belgium.

Berntsen, J., Espelid, T. O., and Genz, A. (1991), "Algorithm 698: DCUHRE—An Adaptive Multidimensional Integration Routine for a Vector of Integrals," *ACM Transactions on Mathematical Software*, 17, 452–456.

Böckenholt, U. (1992), "Thurstonian Representation for Partial Ranking Data," *British Journal of Mathematical and Statistical Psychology*, 45, 31–49.

Bukszár, J., and Prékopa, A. (2001), "Probability Bounds With Cherry Trees," *Mathematics of Operations Research*, 26, 174–192.

Bukszár, J., and Szántai, T. (1999), "Probability Bounds Given by Hyper-Cherry Trees," *Alkalmazott Matematikai Lapok*, 19, 69–85. (in Hungarian); (English translation to appear in *Optimization Methods and Software.*)

Deák, I. (1980), "Three Digit Accurate Multiple Normal Probabilities," *Numerische Mathematik*, 35, 369–380.

——— (1990), *Random Number Generators and Simulation*, Budapest: Akadémiai Kiadó.

——— (in press), "Probabilities of Simple $n$-dimensional Sets for the Normal Distribution," *IIE Transactions.*

Donnelly, T. G. (1973) "Algorithm 462: Bivariate Normal Distribution," *Communications of the ACM*, 16, 638.

Drezner, Z., and Wesolowsky, G. O. (1990), "On the Computation of the Bivariate Normal Integral," *Journal of Statistical Computation and Simulation*, 35, 101–107.

Gassmann, H. (1988), "Conditional Probability and Conditional Expectation of a Random Vector," in *Numerical Techniques for Stochastic Optimization*, eds. Yu. Ermoliev, and R.J-B Wets, Springer Series in Computational Mathematics, Vol. 10, New York: Springer Verlag.

Genz, A. (1992), "Numerical Computation of Multivariate Normal Probabilities," *Journal of Computational and Graphical Statistics*, 1, 141–149.

——— (1993), "Comparison of Methods for the Computation of Multivariate Normal Probabilities," *Computing Science and Statistics*, 25, 400–405.

——— (1998), Personal homepage, http://www.sci.wsu.edu/math/faculty/genz/homepage.

Gupta, S. S., (1963), "Probability Integrals of Multivariate Normal and Multivariate $t$," *Annals of Mathematical Statistics*, 34, 792–828.

Hajivassilou, V., McFadden, D., and Ruud, P. (1996), "Simulation of Multivariate Normal Rectangle Probabilities: Theoretical and Computational Results," *Journal of Econometrics*, 72, 85–134.

Harris, B., and Soms, A. P. (1980), "The Use of the Tetrachoric Series for Evaluating Multivariate Normal Probabilities," *Journal of Multivariate Analysis*, 10, 252–267.

Hart, J. F., Cheney, E. W., Lawson, C. L., Maehly, H. J., Mesztenyi, C. K., Rice, J. R., Thatcher, Jr., H. G., and Witzgall, C. (1968), *Computer Approximations*, New York: Wiley.

Hill, I. D., and Joyce, S. A. (1967), "Algorithm 304: Normal Curve Integral," *Communications of the ACM*, 10, 374.

Hunter, D. (1976), "Bounds for the Probability of a Union," *Journal of Applied Probability*, 13, 597–603.

Joe, H. (1995), "Approximations to Multivariate Normal Rectangle Probabilities Based on Conditional Expectations," *Journal of the American Statistical Association*, 90, 957–964.

Keast, P. (1973), "Optimal Parameters for Multidimensional Integration," *SIAM Journal on Numerical Analysis*, 10, 831–838.

Kendall, M.G. (1941), "Proof of Relations Connected With the Tetrachoric Series and its Generalizations," *Biometrika*, 32, 196–198.

Korobov, N. M. (1959), "On Approximate Calculations of Multiple Integrals," *Doklady Akademii Nauk SSSR*, 124, 1207–1210, (in Russian).

——— (1960), "Properties and Calculation of Optimal Coefficients," *Soviet Mathematical Doklady*, 1, 696–700.

Marsaglia, G., and Olkin, I. (1984), "Generating Correlation Matrices," *SIAM Journal of Scientific and Statistical Computing*, 5, 470–475.

Ochi, Y., and Prentice, R. L. (1984), "Likelihood Inference in a Correlated Probit Regression Model," *Biometrika*, 71, 531–543.

Prékopa, A. (1974), "Programming Under Probabilistic Constraints with a Random Technology Matrix," *Mathematische Operationsforschung und Statistik*, 5, 109–116.

——— (1995), *Stochastic Programming*, London: Kluwer Academic Publishers.

Prékopa, A., Deák, I., Ganczer, S., and Patyi, K. (1980), "The STABIL Stochastic Programming Model and its Experimental Application to the Electrical Energy Sector of the Hungarian Economy," in *Stochastic programming, Proceedings of the International Conference Oxford 1974*, ed. M. A. H. Dempster, Oxford: Oxford University Press, pp. 369–385.

Prékopa, A., and Szántai, T. (1978), "Flood Control Reservoir System Design Using Stochastic Programming," *Mathematical Programming Study*, 9, 138–151.

Saltykov, A. I. (1963), "Tables for the Computation of Multiple Integrals Using the Method of Optimal Coefficients," *USSR Computational Mathematics and Mathematical Physics*, 3, 181–186.

Schervish, M. J. (1984), "Multivariate Normal Probabilities With Error Bound," *Applied Statistics*, 33, 81–87.

Somerville, P. N. (1998), "Numerical Computation of Multivariate Normal and Multivariate $t$ Probabilities Over Convex Regions," *Journal of Computational and Graphical Statistics*, 7, 529–544.

Szántai, T. (1976), "An Algorithm for Calculating Values and Gradient Vectors for the Multiple Normal Distribution Function," *Alkalmazott Matematikai Lapok*, 2, 27–39 (in Hungarian).

——— (1986), "Evaluation of a Special Multivariate Gamma Distribution Function," *Mathematical Programming Study*, 27, 1–16.

——— (2000), "Improved Bounds and Simulation Procedures on the Value of the Multivariate Normal Probability Distribution Function," *Annals of Operations Research*, 100, 85–101.

Tomescu, I. (1986), "Hypertrees and Bonferroni Inequalities," *Journal of Combinatorial Theory*, Series B, 41, 209–217.

Tong, Y. L. (1990), *The Multivariate Normal Distribution*, New York: Springer Verlag.

Vijverberg, W. P. M. (1997), "Monte Carlo Evaluation of Multivariate Normal Probabilities," *Journal of Econometrics*, 76, 281–307.

Worsley, K. J. (1982), "An Improved Bonferroni Inequality and Applications," *Biometrika*, 69, 297–302.