To appear: Communications in Statistics: Simulation and Computation 23, 3, (1994)

THE TRANSFORMED REJECTION METHOD FOR GENERATING RANDOM VARIABLES, AN ALTERNATIVE TO THE RATIO OF UNIFORMS METHOD

Wolfgang Hörmann and Gerhard Derflinger

University of Business Administration Vienna
Institute f. Statistics
Augasse 2-6, A-1090 Vienna, AUSTRIA

Key Words and Phrases: *random variate generation; transformed rejection method; normal distribution; t-distribution.*

ABSTRACT

Theoretical considerations and empirical results show that the one-dimensional quality of non-uniform random numbers is bad and the discrepancy is high when they are generated by the ratio of uniforms method combined with linear congruential generators. This observation motivates the suggestion to replace the ratio of uniforms method by transformed rejection (also called exact approximation or almost exact inversion), as the above problem does not occur for this method. Using the function $G(x) = \left(\frac{a}{1-x} + b\right) x$ with appropriate a and b as approximation of the inverse distribution function the transformed rejection method can be used for the same distributions as the ratio of uniforms method. The resulting algorithms for the normal, the exponential and the t-distribution are short and easy to implement. Looking at the number of uniform deviates required, at the code length and at the speed the suggested algorithms are superior to the ratio of uniforms method and compare well with other algorithms suggested in literature.

1 INTRODUCTION

The ratio of uniforms method, suggested in (Kinderman and Monahan 1977), has become a popular and frequently used method for generating non-uniform random numbers. To obtain random numbers from a distribution with density function f a point (U, V) is generated from the uniform distribution over the region $C_h = \left\{(u, v) : 0 \leq u \leq \sqrt{h(v/u)}\right\}$ where h is proportional to f. Then $X = V/U$ has the desired distribution. To generate the point (U, V) over C_h rejection from the minimal rectangle enclosing C_h is used in most cases. This rectangle exists for all bounded densities f with subquadratic tails, which shows one advantage of the method: It is flexible and thus applicable to many different distributions such as the normal, exponential, gamma, beta, Student-t, etc. The resulting algorithms are simple but nevertheless quite fast. Using appropriate functions, so-called squeezes, to avoid the calculation of $\sqrt{h(v/u)}$ in most cases, yields fast algorithms with only two additional lines of code.

On the other hand the ratio of uniforms method has some disadvantages: The acceptance probability is quite low (for none of the standard distributions mentioned above it is greater than 80%) and thus the expected number of uniform deviates required is high as well (greater than 2.5 for the above examples). The third and most important disadvantage concerns the quality of the ratio of uniforms method when a linear congruential generator (LCG) is used as the source of uniform random numbers. In (Hörmann 1994a) and (Hörmann 1994b) theoretical considerations and empirical calculations show that the quality of the ratio of uniforms method combined with a LCG is much lower than the quality of the LCG itself. On the other hand the large number of uniform deviates required makes the use of the ratio of uniforms method together with a slower uniform number generator with a better two-dimensional distribution quite slow. So it seems justified to think about replacing the ratio of uniforms method by a different transformation method without the mentioned disadvantages.

As the ratio of uniforms method with rectangles can be viewed as rejection from a table mountain distribution with density function $f(x) = \frac{1}{4} \min(1, 1/x^2)$ it is an obvious idea to replace the ratio of uniforms method by a rejection method with $f(x)$ as dominating density. The computations of (Afflerbach and

Hörmann 1992) show that the quality of the table mountain rejection (where the table mountain density is generated by inversion) combined with a LCG is much better than that of the ratio of uniforms method. On the other hand the table mountain distribution does not fit very well in most cases. So we looked for a different class of distributions that can be easily generated by inversion and yield high acceptance probabilities when used as dominating density for a rejection algorithm. The best results were obtained for the inverse cumulative distribution function $G(x) = \left(\frac{a}{1-x} + b\right) x$ for positive random variables and $G(x) = \left(\frac{2a}{1/2-|x-1/2|} + b\right) (x - 1/2)$ for symmetric ones. We used the method of *transformed rejection* as already described in (Wallace 1976). In (Marsaglia 1984) almost the same method is called *exact-approximation*, in (Devroye 1986) it is called *almost-exact inversion*. A straight forward calculation shows that the tails of the density function associated with the above G are $O(1/x^2)$ like the tails of the table-mountain. Therefore transformed rejection with the above function G is applicable to all distributions that can be generated with the ratio of uniforms method. For the Poisson, the binomial and the normal distribution see (Hörmann 1993c), (Hörmann 1993a) and (Hörmann 1993b). Transformed rejection has the advantages of the ratio of uniforms method (i.e. short and fast algorithms) but does not have some of its disadvantages. The acceptance probability for this method is considerably higher, the number of uniform deviates required can be reduced far below two using decomposition techniques and the quality when combined with a LCG is close to the quality of the LCG itself. The details of the method and the assertions concerning the advantages will be clarified in the following sections.

2 TRANSFORMED REJECTION

If we use the rejection method to generate random numbers with density function f we need a dominating density or hat function h and a real number α with $f(x) \leq h(x)/\alpha \, \forall x$. Then we generate a random number X from the dominating density and a uniform random number V. If $V \leq \alpha f(X)/h(X)$

then X is accepted as a random number from the density f, otherwise X is rejected and the procedure starts again. The acceptance-probability is α.

For the transformed rejection method we start with the inverse distribution function $G(u)$ $(0 \leq u \leq 1)$ of the dominating distribution. (Random numbers of this distribution are of course generated by inversion.) As the dominating density is $(G^{-1})'(x)$ the acceptance condition now reads:

$$V \leq \frac{\alpha f(X)}{(G^{-1})'(X)}$$

In many cases it is too difficult to calculate the dominating density even for very simple G. But this is not necessary. As $(G^{-1})'(x) = 1/G'(u)$ for $x = G(u)$ the acceptance condition can be transformed into:

$$V \leq \alpha f(G(U))G'(U)$$

Now we can give the basic algorithm for transformed rejection.

Algorithm Transformed Rejection (TR):

1: Generate two uniform random numbers U and V.

2: If $V \leq \alpha f(G(U))G'(U)$ return $G(U)$, else go to 1.

In Marsaglia's paper (Marsaglia 1984) the same method is presented under a different point of view which is based on its relation to the inversion method. To generate random numbers with density f take a function G which is close to the inverse distribution function corresponding to f. Then use rejection to generate random numbers X with density $f(G(x))G'(x)$ and return $G(X)$. If G is close to the inverse distribution function of the desired distribution then $f(G(x))G'(x)$ is close to the density of the (0,1) uniform distribution. Explained in this way the method is called exact approximation in (Marsaglia 1984) and almost exact inversion in (Devroye 1986). If G is chosen properly there is a large rectangle between the curve $\alpha f(G(u))G'(u)$ and the u-axis. Figure I shows $\alpha f(G(u))G'(u)$ together with that rectangle for the exponential distribution. Figure II shows the same curves as Figure I but in the scale of an ordinary rejection algorithm with hat function, density function scaled down with α, and squeeze. We obtain Figure II by transforming the pair (U, V) of Figure I into $(G(U), V/G'(U))$. Figure III shows the curve $\alpha f(G(u))G'(u)$ together with the rectangle for the normal, Figure IV for the Cauchy distribution.

Figure I

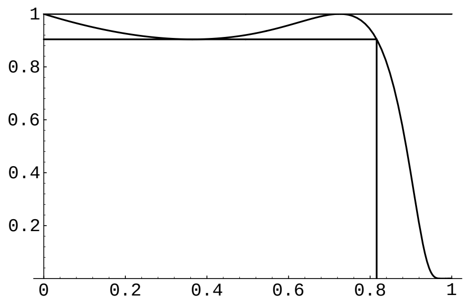


Figure II

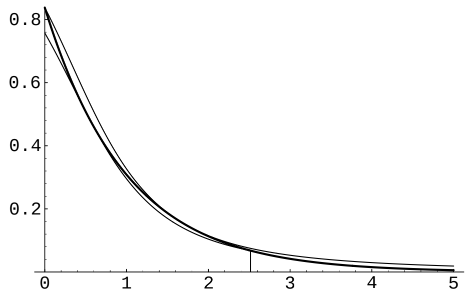


Figure III

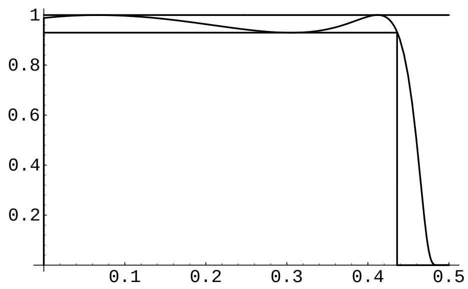
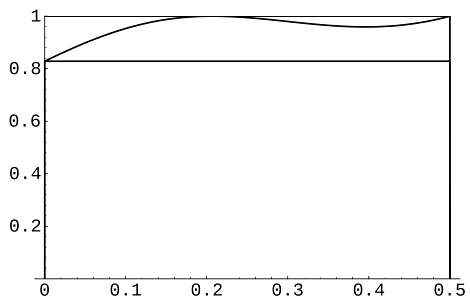


Figure IV



As most of the time in the transformed rejection method is spent in evaluating the acceptance condition this rectangle can be used as a squeeze function to accelerate the algorithm. The following algorithm is an example for the use of squeezes if the rectangle is $(0, u_r) \times (0, v_r)$ (which will occur for monotone densities).

Algorithm Transformed Rejection with Squeeze (TRS):

1: Generate two uniform random numbers U and V.

2: If $U \leq u_r$ and $V \leq v_r$ return $G(U)$.

3: If $V \leq \alpha f(G(U))G'(U)$ return $G(U)$, else go to 1.

Using the idea of decomposition (cf. for example (Devroye 1986)) the expected number of uniform random numbers required can be reduced as well, which yields the following refinement of Algorithm TRS.

Algorithm Transformed Rejection with Decomposition (TRD):

1: Generate a uniform random number V.

2: If $V \leq u_r v_r$ return $G(V/v_r)$.

3: If $V \geq v_r$ generate a uniform random number U,

4: else set $U \leftarrow V/v_r$, generate a uniform random number V in $(0, v_r)$.

5: If $V \leq \alpha f(G(U))G'(U)$ return $G(U)$, else go to 1.

For symmetric density functions it is more convenient to define the transformation G on the interval (-0.5,0.5) instead of on (0,1). The rectangle below the curve will be denoted by $(-u_r/2, u_r/2) \times (0, v_r)$. The only change necessary in Algorithm TR is to generate the uniform random number U in step 1 in (-0.5,0.5) instead of (0,1). The same change is necessary in Algorithm TRS. In addition $U \leq u_r$ of step 2 must be replaced by $|U| \leq u_r/2$. As the changes of Algorithm TRD are more complicated we give the whole Algorithm:

Algorithm TRD for symmetric densities (TRDs):

1: Generate a uniform random number V.

2: If $V \leq u_r v_r$ return $G(V/v_r - u_r/2)$.

3: If $V \geq v_r$ generate a uniform random number U in (-0.5,0.5),

4: else set $U \leftarrow V/v_r - (u_r+1)/2$, $U \leftarrow \text{sign}(U)0.5 - U$, generate a uniform random number V in $(0, v_r)$.

5: If $V \leq \alpha f(G(U))G'(U)$ return $G(U)$, else go to 1.

The expected number of uniform deviates needed for one non-uniform random number is $2/\alpha$ for Algorithms TR and TRS. By decomposition the number of uniform deviates required is reduced to $(2 - u_r v_r)/\alpha$ for Algorithm TRD. The expected number of evaluations of the acceptance condition necessary is $1/\alpha$ for Algorithm TR and $(1 - u_r v_r)/\alpha$ for Algorithms TRS and TRD .

3 EXAMPLES

The class $G(x) = \left(\frac{a}{1-x} + b\right) x$, $0 \leq x < 1$, and its symmetric version $G(x) = \left(\frac{2a}{1/2-|x|} + b\right) x$, $-0.5 < x < 0.5$, turned out to yield high acceptance probabilities for a variety of distributions. As $G(x)$ can be computed very fast as well, we restricted ourselves to this class. To design algorithms for special densities which are monotone or symmetric it is necessary to chose good values for a and b for the respective G in a way that the curve $f(G(U))G'(U)$ is as close to one as possible. We tried to chose a and b such that α is maximized or equivalently the maximum of $f(G(U))G'(U)$ (i.e. $1/\alpha$) is minimized. As a solution in closed form was impossible we used a numerical optimization procedure in two stages: first to find the value for $1/\alpha$ for the different fixed values of a and b and then to optimize $1/\alpha$ by varying a and b. In Mathematica (or a similar mathematical package) this optimization can be implemented in a simple function with less than five lines of code. Reasonable starting values can be taken from the values for the standard distributions contained in the Tables I to V. If a and b are chosen $1/\alpha$ is computed as the maximum of $f(G(U))G'(U)$ and u_r and v_r are the sides of the rectangle below $\alpha f(G(U))G'(U)$. For the case of distributions that are almost symmetric it is possible to proceed as in the symmetric case as it was done for the Poisson and binomial distributions in (Hörmann 1993c) and (Hörmann 1993a). If a distribution is strongly asymmetric it is best to optimize the left and the right part separately.

For the normal distribution the transformed rejection method with that G was already suggested in (Wallace 1976) but there the choice of a and b does not maximize α and the improvements of Algorithms TRS and TRD are not mentioned. This seems to be the only reason that the transformed rejection method has not often been used to generate Gaussian random numbers though the relation between speed and code length is in our opinion better than for any other method. (See Table VI and Table XIII.) Table I contains everything necessary to implement the algorithms of section 2 for the normal distribution. For the Cauchy distribution the information is given in Table II. The use of transformed rejection with a similar G to sample from the Cauchy distribution was already suggested in (Ahrens and Dieter 1988).

Table I: Normal distribution

$G(x)$	$(2a/(1/2-\|x\|)+b)\,x$
a	0.062794
b	2.530885
α	0.8904302215
u_r	$2 \cdot 0.4359971734$
v_r	0.9296123611
Acceptance-condition	$\left(V \cdot \exp(G(U)^2/2) - \alpha b/\sqrt{2\pi}\right)(1/2-\|U\|)^2 \leq \alpha a/\sqrt{2\pi}$

Table II: Cauchy distribution

$G(x)$	$(2a/(1/2-\|x\|)+b)\,x$
a	0.306327
b	1.479078
α	0.9623546527
u_r	$2 \cdot 0.5$
v_r	0.8284264502
Acceptance condition	$(V(1+x^2) - \alpha b/\pi)\,(1/2-\|U\|)^2 \leq \alpha a/\pi$

A method that works well for the normal and for the Cauchy distribution is applicable for the t-distribution with ν degrees of freedom, $\nu \geq 1$, as well. It is only necessary to determine suitable values for the parameters of G in dependence of ν. One possibility is to approximate the optimal values of a, b, u_r, v_r and $\alpha \cdot const$ (where *const* denotes the normalization constant of the t-distribution) for different ν by functions. The result of this work is given in Table III. Using Algorithm TRSD and Table 3 we obtain a generator for the t-distribution ($\nu \geq 1$) which is very fast if ν does not change often but has a slow setup. If ν is changing frequently it is better to divide the range of ν $[1, \infty)$ into 8 intervals and to use fixed values within these intervals which are stored in arrays. The setup is then reduced to finding the number of the correct interval and thus is very fast. The values of the parameters for the eight intervals are given in Table IV.

Table III: t-Distribution with ν degrees of freedom

$G(x)$	$(2a/(1/2-\|x\|)+b)\,x$
a	$0.062794+\frac{7}{30}\nu^{-1.35}$
b	$2.530885-\nu^{-1.75}$
$\alpha\cdot const$	$0.036162\cdot b+0.252453+$ $c_{[3,\infty)}(\nu)\left(0.0104466\exp\left(-\frac{7.04}{\nu-2.5}\right)\right)+$ $c_{[1,1.0261)}(\nu)\left(-0.011686+(\nu-1)(11.427-10.7\nu)\right)$
$u_r/2$	$c_{[1.4346,\infty)}(\nu)\left(0.4375+\frac{0.198}{\nu-0.372}-\frac{0.252}{\nu^{1.196}}\right)+$ $c_{[1,1.4346)}(\nu)\left(0.5-0.09137(\nu-1)\right)$
v_r	$c_{[1.4346,\infty)}0.91697773+c_{[1,1.4346)}(\nu)\left(0.5444+0.2597\nu\right)$
Acceptance-*condition*	$\left(\frac{\nu+1}{2}\right)\log\left(1+\frac{G(U)^2}{\nu}\right)\le\log\left(\alpha\cdot const\left(\frac{a}{(1/2-\|U\|)^2}+b\right)/V\right)$

$c_{(a,b)}$ denotes the characteristic function of the interval (a,b).

Table IV: t-Distribution

degrees of freedom	a	b	$\alpha\cdot const$	u_r	v_r
$(1\|1.23)$	0.3	1.6	0.3	$2\cdot 0.4324$	0.82
$(1.23\|1.7)$	0.21	2.12	0.31279	$2\cdot 0.4194$	0.85
$(1.7\|2.5)$	0.17	2.15	0.32655	$2\cdot 0.4026$	0.9241
$(2.5\|4)$	0.13	2.325	0.33561	$2\cdot 0.3970$	0.9496
$(4\|8)$	0.105	2.406	0.34237	$2\cdot 0.4015$	0.9496
$(8\|19)$	0.08	2.495	0.34843	$2\cdot 0.4209$	0.9324
$(19\|60)$	0.073	2.5	0.35219	$2\cdot 0.4238$	0.9403
$(60\|\infty)$	0.063	2.537	0.35401	$2\cdot 0.4357$	0.9228

An example for a positive distribution that can be generated with the transformed rejection method is the exponential distribution. The details are contained in Table V. It is also possible to use the transformed rejection method with the suggested G to generate random numbers of the Gamma distribution with shape parameter $a \geq 1$. The resulting acceptance probabilities lie between 83% for $a = 1$ and 89% when $a \to \infty$). But the work to get a setup for changing parameter a was not done yet.

Table V: Exponential distribution

$G(x)$	$(a/(1-x)+b)\,x$
a	0.426
b	0.7675
α	0.8378998
u_r	0.816005087
v_r	0.9040791868
Acceptance condition	$(V \exp(G(U)) - \alpha b)\,(1-U)^2 \leq \alpha a$

4 COMPARISON OF THE ALGORITHMS

The first and most important motivation to look for a method that can replace the ratio of uniforms method were problems with the quality of random variates generated by the ratio of uniforms method in combination with a linear congruential uniform random number generator (LCG), first investigated in (Afflerbach and Hörmann 1992). The detailed discussion in (Hörmann 1994a) is summarized as follows: The ratio of uniforms method transforms a pair of uniform random numbers into one non-uniform number. It is obvious, that pairs lying on the same line through the origin are transformed into the same non-uniform random number. Together with the lattice structure of the pairs returned by a LCG this implies that there is always a relatively large gap without a pseudo-random number in the direction of the shortest lattice vector. In (Hörmann 1994a) it is proven that the probability of this gap is of the order $1/\sqrt{m}$ where m denotes the modulus of the LCG. Therefore the one-dimensional resolution of the ratio of uniforms method is small compared with the resolution of the underlying LCG. In (Hörmann 1994b) a statistical test based on the sum of squares of spacings is described that detects this bad resolution of the ratio of uniforms method in samples of size $3 \cdot 10^6$. These results really indicate that the use of ratio of uniforms in combination with a LCG can well influence the results of a large scale simulation and should not be used any longer.

On the other hand it is shown in (Hörmann 1994a) that there are no problems with the quality of the ratio of uniforms method if it is combined with a multiple recursive linear congruential generator (MRLCG). So we compare some characteristics of ratio of uniforms and transformed rejection algorithms for the normal, t, Cauchy and exponential distributions. For the t-distribution we used the table-version of the transformed rejection algorithms based on the results of Table IV. For the normal distribution we include the well-known sine/cosine (or Box-Muller) method BM, the ACR method (Hörmann and Derflinger 1990) and algorithm KR (Kinderman and Ramage 1976). For random variate generation algorithms simplicity is of great importance especially as parts of the program that are executed with very low probability are extremely difficult to debug. As a crude measure for this simplicity Table VI contains the number of C-statements of our implementation without function and variable declarations. A second important characteristic, the expected number of uniform deviates required to generate one non-uniform variate, is given in Table VII. Table VIII contains the average execution time for our C-implementations of the different methods on our DECstation 5000/240 using a MRLCG with $m = 2^{31} - 1$ which takes 2.8 μ-seconds. The third line of the t-distribution gives the average generation time when ν varies with probability 0.5 in the given interval.

Table VI: Number of C-statements

	TRS	TRD	RoU	BM	ACR	KR
Normal distribution	9	14	10	10	34	60
t-distribution	12	27	19			
Cauchy distribution	9	10	8			
exponential distribution	8	13	9			

Table VII: Expected number of uniform deviates required

	TRS	TRD	RoU	BM	ACR	KR
Normal distribution	2.246	1.336	2.738	1	1.485	2.16
t-distribution $\nu = 20$	2.246	1.339	2.704			
t-distribution $\nu = 3$	2.170	1.285	2.546			
Cauchy distribution	2.078	1.217	2.546			
exponential distribution	2.387	1.506	2.943			

Table VIII: Execution time in μ-seconds

	TRS	TRD	RoU	BM	ACR	KR
Normal distribution	8.6	6.3	10.4	9.4	5.8	7.3
t-distribution $\nu = 20$	10.0	8.0	11.2			
t-distribution $\nu = 3$	10.0	8.3	10.7			
t-dist. $1 \leq \nu \leq 100$	14.5	12.9	50.8			
Cauchy distribution	7.5	5.2	8.6			
exponential distribution	9.3	6.9	12.0			

Table VI shows that Algorithm TRS is about as simple as ratio of uniforms, TRD is only slightly longer. Nevertheless the results of Table VII indicate that, due to the high acceptance probability, Algorithm TRS needs considerably less uniform random numbers than the ratio of uniforms method for any of the distributions. For Algorithm TRD this number is reduced to values between 1.21 and 1.51, which is really low, especially for the t-distribution where no algorithms that need less than two uniform deviates were suggested in literature (cf. (Stadlober and Dieter 1990) and (Kinderman and Monahan 1980)). The timing results show that TRS is faster than the corresponding ratio of uniforms algorithm, TRD is faster than TRS. Compared with the fastest algorithms for the respective distribution Algorithm TRD is not competitive for the exponential and the Cauchy distributions. For the normal distribution TRD is faster than KR and only slightly slower than ACR but has much shorter code. For the t-distribution algorithm TRS is very short but as fast as any other method (we compared it with all methods described (Kinderman and Monahan 1980) and (Stadlober and Dieter 1990)) the execution time is almost the same for all values of ν and the setup time is very low. Algorithm TRD is again longer and faster.

5 CONCLUSION

It has been demonstrated that the transformed rejection method with the suggested transformation G yields fast and short algorithms to sample from various distributions, which are easy to implement. Compared with the ratio-of-uniforms method the acceptance probabilities are higher. Because of the bad one-dimensional distribution of the ratio of uniforms method when combined with LCG's it should be replaced by transformed rejection combined with a LCG with a multiplier large compared with $\sqrt{m}$. When uniform random numbers with a better two-dimensional distribution are used (eg. a multiple recursive congruential generator) the lower speed of the uniform generator and the lower amount of uniform random numbers required by Algorithms TRS and TRD suggest the use of the transformed rejection method as well. For the t-distribution the proposed algorithms are shorter and faster than the best algorithms presented in literature.

BIBLIOGRAPHY

Afflerbach, L. and Hörmann, W. 1992. Nonuniform random numbers: a sensitivity analysis for transformation methods. *In:* Dieter, U. and Pflug, G.Ch. (eds), *International Workshop on Computationally Intensive Methods in Simulation and Optimization, Lecture Notes in Econom. Math. Systems 374*. Berlin: Springer-Verlag.

Ahrens, J. H. and Dieter, U. 1988. Efficient table-free sampling methods for the exponential, Cauchy, and normal distributions. *Communications of the ACM*, **31**, 1330–1337.

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

Hörmann, W. 1993a. The generation of binomial random variates. *Journal of Statistical Computation and Simulation*, **46**, 101–110.

Hörmann, W. 1993b. New generators of normal and poisson deviates based on the transformed rejection method. *Pages 334–341 of:* K.-W., Hansmann, *et al.* (eds), *Operations Research Proceedings 1992.* Berlin: Springer Verlag.

Hörmann, W. 1993c. The transformed rejection method for generating Poisson random variables. *Insurance: Mathematics and Economics*, **12**, 39–45.

Hörmann, W. 1994a. A note on the quality of random variates generated by the ratio of uniforms method. *ACM Transactions on Modelling and Computer Simulation*, (to appear).

Hörmann, W. 1994b. The quality of non-uniform random numbers. *In:* Dyckhoff, H. et al. (ed), *Operations Research Proceedings 1993.* Berlin: Springer Verlag. (to appear).

Hörmann, W. and Derflinger, G. 1990. The ACR method for generating normal random variables. *OR Spektrum*, **12**, 181–185.

Kinderman, A. J. and Monahan, F. J. 1977. Computer generation of random variables using the ratio of uniform deviates. *ACM Trans. Math. Soft.*, **3**, 257–260.

Kinderman, A. J. and Monahan, J. F. 1980. New methods for generating student's t and gamma variables. *Computing*, **25**, 369–377.

Kinderman, A. J. and Ramage, J. G. 1976. Computer generation of normal random variables. *Journal of the American Statistical Association*, **71**, 893–896.

Marsaglia, G. 1984. The exact–approximation method for generating random variables in a computer. *Journal of the American Statistical Association*, **79**, 218–221.

Stadlober, E. and Dieter, U. 1990. Computer methods for generating student-t variates. *In: Festschrift für Helmut Florian.* Technische Universität Graz, Austria.

Wallace, C. S. 1976. Transformed rejection generators for gamma and normal pseudo-random variables. *Australian Computer Journal*, **8**, 103–105.