NEW GENERATORS OF NORMAL AND POISSON DEVIATES BASED ON THE TRANSFORMED REJECTION METHOD

Wolfgang Hörmann, WU Wien, Augasse 2-6, A-1090

Abstract: The transformed rejection method uses inversion to sample from the dominating density of a rejection algorithm. But in contrast to the usual method it is enough to know the inverse distribution function $F^{-1}(x)$ of the dominating density. This idea can be applied to various continuous (e.g. normal, Cauchy and exponential) and discrete (e.g. binomial and Poisson) distributions with high acceptance probabilities. The resulting algorithms are short, simple and fast. Even more important is the fact that the quality of the method when used in combination with a linear congruential uniform generator is high compared with the quality of the ratio of uniforms method. In addition transformed rejection can be easily employed for correlation induction.

Zusammenfassung: Die Transformierte Verwerfung verwendet Inversion um Zufallszahlen aus der "Hutfunktion" eines Verwerfungsalgorithmus zu erzeugen. Anders als bei der gewöhnlichen Verwerfungsmethode genügt es, die inverse Verteilungsfunktion des "Hutes" zu kennen. Verschiedene stetige und diskrete Verteilungen wie Normal-, Cauchy-, Exponential-, Poisson- und Binomialverteilung können mit dieser Methode erzeugt werden. Die entsprechenden Algorithmen sind kurz, einfach und schnell. Noch wichtiger scheint, daß Ihre Qualität in Kombination mit linearen Kongruenzgeneratoren besser ist als die der Quotientenmethode. Die transformierte Verwerfung eignet sich auch gut, um stark korrelierte Zufallszahlen zu erzeugen.

1. Introduction

There are a lot of papers in literature dealing with the generation of independent non-uniform random deviates but unfortunately most of them are mainly concerned with the execution times of the proposed algorithms which are in most cases measured on only one computer in combination with only one uniform random number generator. In some papers the average number of uniform random numbers required to deliver one random deviate and the time complexity of different algorithms are compared whereas questions dealing with the quality of algorithms in combination with certain uniform random number generators are almost entirely neglected. We are convinced that shortness, simplicity and portability of algorithms are of main importance in random-deviate generation, especially as experience (even with published algorithms) shows that it is extremely difficult to find mistakes with small influence on the distribution. Therefore we think it justified to analyze the properties of a simple method called transformed rejection. The basic idea of it was already suggested in [15]. In the following sections we will demonstrate the many nice properties of transformed rejection and we hope that this will help to increase the use of this method in applications.

2. The method

Transformed rejection as explained in [15] is a combination of the acceptance-rejection and of the inversion principle. 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; X is accepted with probability α.
For the transformed rejection method we start with the inverse distribution function $G(u) (0 \leq u \leq 1)$ of the dominating distribution, i.e. $G(u) = H^{-1}(u)$ where $H(x) = \int_{-\infty}^{x} h(t)\ dt$. To sample from this distribution we simply generate a uniform random number U and take $X = G(U)$. As the dominating density h at point X is $h(X) = (G^{-1})'(X) = 1/G'(U)$ the acceptance condition can be transformed into $V \leq \alpha f(G(U))G'(U)$ and the evaluation of the dominating density is not necessary any longer. Now we can give the basic algorithm for transformed rejection (a reformulation of the algorithm in [15]).

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.

The same method can be 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; 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 [13] and "almost-exact inversion" in [6].
If G is chosen properly there is a large rectangle between the curve $\alpha f(G(u))G'(u)$ and the u-axis. As the evaluation of the acceptance condition is slow the rectangle can be used as a squeeze function to avoid that computation with high probability. In the sequel we shall restrict ourselves to the case that G is symmetric in order to avoid repetitions. For this case it is more convenient to define G on the interval (-0.5,0.5); (the modifications for asymetric G are straightforward). The following algorithm utilizes the idea of squeezes, $(-u_r, u_r) \times (0, v_r)$ denotes a rectangle below the curve $\alpha f(G(u))G'(u)$.

Algorithm Transformed Rejection with Squeeze (TRS):

1: Generate two uniform random numbers U and V, set $U \leftarrow U - 0.5$.

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.

The expected number of evaluations of the acceptance condition necessary to deliver one non-uniform deviate is $1/\alpha$ for Algorithm TR and is reduced to $(1-2u_r v_r)/\alpha$ for Algorithm TRS whereas the expected number of uniform random numbers required (called N in the sequel) remains $2/\alpha$ for both algorithms. To reduce N to $(2-2u_r v_r)/\alpha$ we decompose the quadrangle $(-0.5, 0.5) \times (0, 1)$ into 4 rectangles and use the idea of decomposition together with the "recycling" of uniform random numbers (cf. for example [6]) and obtain:

Algorithm Transformed Rejection with Decomposition (TRD):

1: Generate a uniform random number V. If $V \leq 2u_r v_r$ return $G(V/v_r - u_r)$.

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

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

3. Application to the normal and the Poisson distribution

The transformed rejection method seems to be applicable to any continuous distribution. To use it for discrete distributions it is only necessary to replace the density function by the histogramm and to return $\lfloor G(\cdots) \rfloor$ instead of $G(\cdots)$ in the above algorithms. The choice of G of course depends on the desired distribution; the class $G(u) = \left(\frac{2a}{1/2-|u|} + b\right) u + c$ and $G'(u) = \frac{a}{(1/2-|u|)^2} + b$ (which was first suggested in [15] for the normal distribution) can be used for any distribution that has a bounded density function with subquadratic tails. It is very flexible and yields high acceptance probabilities for a variety of distributions like normal, t-, gamma, Poisson and the binomial distribution (cf. [8] and [9]). As an illustration we will give the details for the normal and the Poisson distribution. In the next section we will compare the transformed rejection algorithms for these two distributions with algorithms suggested in literature.

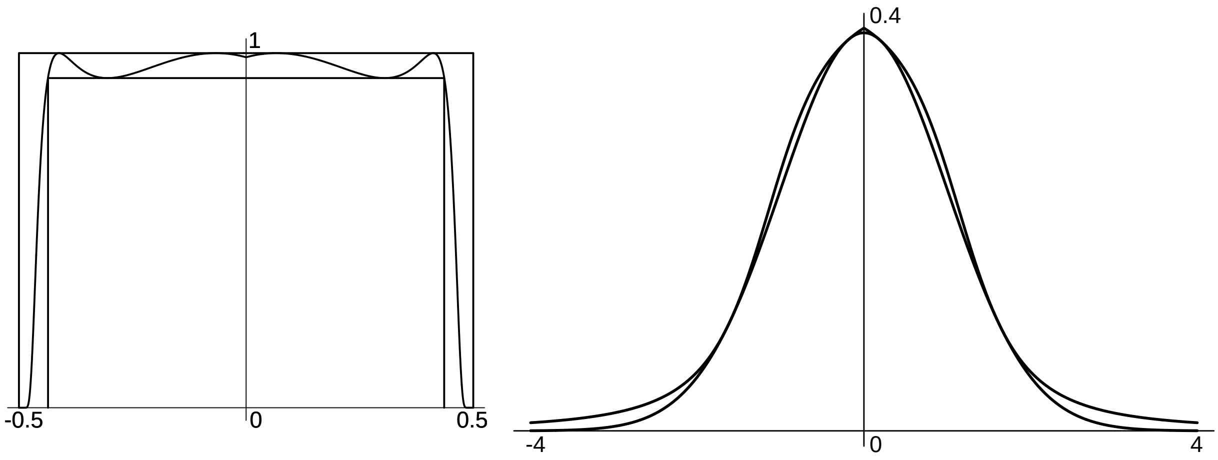
To apply the above algorithms to the normal distribution it is necessary to determine the optimal values for the parameters (the values of a and b given in [15] are not the best possible), which was done by numerical search. Table 1 contains everything necessary to implement the algorithms of above for the normal distribution.

Table 1: Normal distribution

$G(x)$	$\left(\frac{2a}{1/2-\lvert x\rvert} + b\right) x$	
$a = 0.062794$	$b = 2.530885$	$c = 0$
$\alpha = 0.8904302215$	$u_r = 0.4359971734$	$v_r = 0.9296123611$
Acceptance condition	$\left(V \cdot \exp(G(U)^2/2) - \alpha b/\sqrt{2\pi}\right)(1/2 - \lvert U\rvert)^2 \leq \alpha a/\sqrt{2\pi}$	

To show the good fit of the hat function Figure 1 contains the curve $\alpha f(G(U))G'(U)$ together with the rectangle $(-u_r, u_r) \times (0, v_r)$ on the left hand side, the density function with the hat function asociated with $G(u)$ on the right hand side, both for the normal distribution.

Figure 1



For the Poisson distribution it is not difficult to compute a table similar to Table 1 for a fixed value of the Poisson-parameter μ. But in most applications we need a Poisson random number generator that works for arbitrary μ. Therefore it is necessary that we can compute the parameters of the transformed rejection algorithm for a certain μ in a short set-up step. So we computed optimal values for a, b and c for many μ by numerical search and approximatexd these optimal values by simple functions of μ. As there are simple and short Poisson generators that are very fast for small μ we decided to approximate the parameters of G only for $\mu \geq 10$. For these approximations of a, b and c (cf. Table 2) we calculated the exact values of u_r, v_r and α and found easy lower bounds depending on μ which are also given in Table 2. The asymptotical validity of these lower bounds can be seen easily, for finite values of μ the validity was checked by extensive numerical calculations.

Table 2: Poisson distribution, approximations valid for $\mu \geq 10$

c	$\mu + 0.445$
b	$0.931 + 2.53\sqrt{\mu}$
a	$-0.059 + 0.02483b$
$1/\alpha$	$1.1239 + 1.1328/(b - 3.4)$
u_r	0.43
v_r	$0.9277 - 3.6224/(b - 2)$
Acceptance condition	$\log(v/\alpha/(a/(0.5 - \lvert u \rvert)^2 + b) \leq -\mu + \lfloor G(u) \rfloor \log(\mu) - \log(\lfloor G(u) \rfloor !)$

The acceptance condition in Table 2 is obtained by taking the logarithm on both sides of the standard inequality. $\log(k!)$ can be evaluated easily in a separate function with help of the Stirling approximation for $k \geq 10$; for $k < 10$ the exact values should be stored in a table. For μ large cancellation can occur when evaluating the acceptance condition but experience has shown that for the suggested algorithm the influence of cancellation is negligible for $10 \leq \mu \leq 10^7$ if 64 bit floating point numbers are used. Of course there are various methods to plug the Stirling approximation directly into the acceptance condition which are slightly faster and numerically a bit more stable (cf. [9]) but as we already stated in the introduction we think that simplicity and portability are more important for standard applications.

4. Comparison with existing algorithms

4.1 Simplicity and speed

Table 3: Normal generators: Execution times in μ-seconds

	reference	DEC A	DEC B	PC-386 A	PC-386 B	number of C-statements	N
NTRS		5.0	8.6	125	260	9	2.25
R.o.U	[12] and [6]	6.0	10.4	142	305	10	2.74
sine-cosine	[3]	7.8	9.5	116	176	10	1
Polar	[6] p. 235	6.0	8.1	113	190	12	1.27
NTRD		4.1	6.3	102	181	14	1.34
ACR	[7]	3.3	5.8	81	171	35	1.49
KR	[11]	3.7	7.3	91	221	60	2.16

Table 3 compares the transformed rejection normal generators NTRS and NTRD with several of the fastest normal generators designed for a high level language implementation (cf. [11] and [7]). All of them were coded in C and tested on a DECstation 5000/240 and on a PC 386/25 with Turbo C. Uniform generator A is the C-implementation of a LCG with modulus 2^{32} taking 0.86 μ-seconds on DEC and 20 on the PC, uniform generator B is a multiple recursive generator taking 2.83 μ-seconds on DEC and 80 on the PC. It is difficult to give a measure for the simplicity of an algorithm, as a crude approximation we used the number of C-statements of our implementations. The results of Table 3 show that the transformed rejection algorithms compare very well with the best existing normal generators if we take both code length and speed into account. In addition Table 3 demonstrates that a short algorithm need not be slow as especially on the PC some of the short methods have practically the same speed as the two longer ones.

For the Poisson distribution the simple and short inversion procedure with search from the origin has one serious drawback: The execution time grows linearly with μ. Therefore rejection algorithms with uniformly bounded execution times were suggested in literature (eg. [2], [5] and [10]) which have values of N much larger than 2 or are very complicated. As in the normal case the transformed rejection algorithms compare very well with these existing algorithms as far as code length, speed of the algorithms and number of uniforms required are concerned. For details see [9].

4.2 Quality

The quality of non-uniform random number generators is difficult to analyse as it depends mainly on the quality of the uniform number generator used. Therefore most authors simply state that the proposed method is "exact" which means that truly uniform and independent random numbers (i.e. "ideal" random numbers which are not available on a computer) are transformed into an independent sample of the desired distribution. Of course this is true for the transformed rejection method as well.
A heuristic statement concerning the quality of different non-uniform random number generators says: A method has good quality if it uses a low amount of uniform random numbers (i.e. N is close to 1). This assertion is supported by the consideration that all uniform pseudo-random numbers have a better resolution in low dimensions than in higher ones. The results of [1] indicate that in combination with linear congruential generators (LCGs) the assertion above is a simplification but not entirely wrong.
Looking at Table 3 we see that N is quite low for Algorithm TRD. This is true not only for the normal distribution but for any application of TRD as long as the transformation G is selected with care. We think that this is an important advantage of the transformed rejection method over competing methods which are as simple and as flexible like ordinary rejection or the ratio of uniforms method. Both methods have no standard trick like Algorithm TRD to reduce N below 2.
In the case that non-uniform random numbers are generated by transforming pseudo-random numbers generated by a LCG it is possible to make some statements about the one-dimensional approximation of the desired distribution. The empirical results of [1], Section 3, lead to one important conclusion: The ratio of uniforms method approximates the one-dimensional distribution much worse than other methods like inversion or rejection; therefore the ratio of uniforms method combined with a LCG should not be used for applications that need a good one-dimensional resolution. This result is especially important for the transformed rejection method as any ratio of uniforms algorithm can be replaced by a transformed rejection algorithm with G as suggested above.

4.3 Correlation induction

In many simulation experiments variance reduction can be obtained by inducing positive (common random numbers) or negative (antithetic variates) correlation between the random deviates generated (see simulation text books e.g. [4]). The highest (or lowest) correlation possible can always be obtained with the inversion method but it is also possible to obtain correlation induction with rejection methods as demonstrated in [10] and [14]. Algorithm TRS is especially well suited to install correlation induction facilities as it is so close to the inversion method. Therefore the monotonicity is already at hand and it is only necessary to establish synchronization by using two random number streams: The first one for the two random numbers necessary for the first acceptance-rejection experiment, the second stream if the first pair was rejected. So we changed our normal and Poisson algorithms of above following the guidlines of [14] but using only two random numbers from the first stream instead of four for one random deviate. The resulting algorithms called NTRSCI and PTRSCI had only six lines of additional code and almost the same speed as the standard algorithms.

To test NTRSCI we generated positive correlated normal-exponential pairs using NTRSCI for the normal and inversion with the appropriate synchronization (generating but not using a second uniform random number) for the exponential distribution. The resulting correlation was close to 0.62 which is about two thirds of 0.9 which is the maximal correlation obtainable for normal-exponential pairs. On the other hand NTRSCI is more than four times faster (and much shorter) than the normal inversion algorithm.

For the Poisson distribution we compared algorithm PTRSCI and the simple inversion method to generate antithetic variates for many values of μ between 15 and 10000. The correlation obtained with PTRSCI was between -0.66 and -0.89 compared with -0.98 to -1.00 for inversion but simple inversion is horrible inefficient for large μ (eg. more than ten times slower than PTRSCI for $\mu = 100$). In a second experiment we tested the ability of PTRSCI to induce positive correlation between Poisson random variables with different μ via common random numbers. The correlations obtained are given in Table 4 (note that for $\mu < 10$ PTRSCI uses the simple inversion algorithm), the theoretical obtainable correlations are all above 0.98. The results show that it is possible to induce high correlation with transformed rejection algorithms, typically between 65 and 90 percent of the highest possible. Whether the higher speed and the simplicity of TRSCI justify the loss of correlation depends on the application.

Table 4: Correlation induced by PTRSCI via common random numbers

μ_2	5	15	50	100	500	5000
$\mu_1 = 5$	1.00	0.62	0.67	0.68	0.70	0.71
	$\mu_1 = 15$	1.00	0.74	0.74	0.73	0.73
		$\mu_1 = 50$	1.00	0.81	0.80	0.80
			$\mu_1 = 100$	1.00	0.83	0.83

5. Conclusion

We investigated the properties of the normal and Poisson transformed rejection algorithms. We are convinced that their very promising properties like simplicity and speed, low number of uniforms required and good correlation induction possibilities demonstrate the advantages of transformed rejection which are not at all restricted to these two examples. Therefore we think that transformed rejection algorithms would deserve a place in all random variate generation libraries.

References

[1] Afflerbach, L. and Hörmann, W.
Nonuniform random numbers: a sensitivity analysis for transformation methods.
in: International Workshop on ... , U. Dieter and G. Ch. Pflug, ed., Lecture Notes in Econom. Math. Systems, Springer-Verlag, New York (1992)

[2] Ahrens, J. H. and Dieter, U.
Computer generation of Poisson deviates from modified normal distributions.
ACM Transactions on Mathematical Software 8, 163-179 (1982)

[3] Box, J. E. P. and Muller, M. E.
A note on the generation of random normal deviates. Ann. Math. Statist. 29, 610-611 (1958)

[4] P. Bratley, B. L. Fox and L. E. Schrage
A Guide to Simulation. Springer-Verlag, New York (1983)

[5] Devroye, L.
The computer generation of Poisson random variables. Computing 26, 197-207 (1981)

[6] Devroye, L.
Non-Uniform Random Variate Generation. Springer-Verlag, New York (1986)

[7] Hörmann, W. and Derflinger, G.
The ACR method for generating normal random variables. OR Spektrum 12, 181-185 (1990)

[8] Hörmann, W. and Derflinger, G.
The transformed rejection method ... , an alternative to the ratio of uniforms method.
Manuskript, Institut f. Statistik, Wirtschaftsuniversität Wien, (1991)

[9] Hörmann, W.
The transformed rejection method for generating Poisson random variables.
Insurance: Mathematics and Economics, (to appear)

[10] Kachitvichyanukul, V., Cheng, S. J. and Schmeiser, B. W.
Fast Poisson and binomial algorithms for correlation induction.
Journal of Statistical Computation and Simulation 29, 17-33 (1988)

[11] Kinderman, A. J. and Ramage, J. G.
Computer generation of normal random variables.
Journal of the American Statistical Association 71, 893-896 (1976)

[12] Kinderman, A. J. and Monahan, F. J.
Computer generation of random variables using the ratio of uniform deviates.
ACM Transactions on Mathematical Software 3, 257-260 (1977)

[13] Marsaglia, G.
The exact-approximation method for generating random variables in a computer.
Journal of the American Statistical Association 79, 218-221 (1984)

[14] Schmeiser, B. and Kachitvichyanukul, V.
Noninverse correlation induction: guidelines for algorithm development.
Journal of Computational and Applied Mathematics 31, 173-180 (1990)

[15] Wallace, C. S.
Transformed rejection generators for gamma and normal pseudo-random variables.
Australian Computer Journal 8, 103-105 (1976)