Automatic Random Variate Generation for Simulation Input

Hörmann, Wolfgang; Leydold, Josef

Published: 01/01/2000

Automatic Random Variate Generation for Simulation Input

Wolfgang Hörmann, Josef Leydold

Department of Applied Statistics and Data Processing
Wirtschaftsuniversität Wien

Preprint Series

Preprint 33
May 2000

http://statmath.wu-wien.ac.at/

AUTOMATIC RANDOM VARIATE GENERATION FOR SIMULATION INPUT

W. Hörmann

Department of Industrial Engineering,
Boğaziçi University Istanbul,
80815 Bebek-Istanbul, Turkey

J. Leydold

Department for Applied Statistics,
University of Economics and BA Vienna,
Augasse 2-6, A-1090 Vienna, Austria, EU

ABSTRACT

We develop and evaluate algorithms for generating random variates for simulation input. One group called automatic, or black-box algorithms can be used to sample from distributions with known density. They are based on the rejection principle. The hat function is generated automatically in a setup step using the idea of transformed density rejection. There the density is transformed into a concave function and the minimum of several tangents is used to construct the hat function. The resulting algorithms are not too complicated and are quite fast. The principle is also applicable to random vectors. A second group of algorithms is presented that generate random variates directly from a given sample by implicitly estimating the unknown distribution. The best of these algorithms are based on the idea of naive resampling plus added noise. These algorithms can be interpreted as sampling from the kernel density estimates. This method can be also applied to random vectors. There it can be interpreted as a mixture of naive resampling and sampling from the multinormal distribution that has the same covariance matrix as the data. The algorithms described in this paper have been implemented in ANSI C in a library called UNURAN which is available via anonymous ftp.

1 INTRODUCTION

Modeling the input distribution and generating random variates from it is an important task for many simulation studies. In this paper we explain two general approaches that can solve most practical questions that arise in simulation studies not only for random variates but also for random vectors. The C-implementation of the algorithms are available via anonymous ftp in a library called UNURAN (Leydold and Hörmann 2000).

For the first approach we assume that somehow (a priori knowledge, educated guess, fitting to data) we have specified the density of the input distribution. There exists a vast literature on generation methods for continuous standard distributions; see, for example, Devroye (1986), Dagpunar (1988) or Gentle (1998). These algorithms are often especially designed for a particular distribution and tailored to the features of each probability density function. The designing goals for these methods are fast generators and/or simple code. However, unless we decide to use a standard distribution, these algorithms can not be used. But during the last decade so called automatic (or black box) algorithms have been developed to avoid the design of special algorithms. They work for large classes of distributions and require no more than a program that can evaluate (a multiple of) the probability density function. Obviously these universal methods need some setup step, in opposition to special generators, e.g., to the Box-Muller method. But then they generate random variates with a fast marginal generation speed.

The second approach we are going to describe in this paper can be used in the case when we cannot (or do not want to) explicitly specify the density of the input distribution but have observations available from the stochastic input of the real system. Then there exist simple methods to generate random variates more or less directly from the observed data (the given sample). The fact that we can easily sample from kernel density estimates (see e.g. Silverman (1986) and Devroye (1986)) seems to be practically unknown among simulation practitioners.

A third possible approach is to choose the input distribution from the Johnson translation system or from Bezier distributions (see e.g. Nelson and Yamnitsky (1998) or Wagner and Wilson (1996)).

This paper is organised as follows. Section 2 describes the main ideas of transformed density rejection, a universal method for generating from one and multi-dimensional continuous distributions. Section 3 explains how we can use kernel density estimation to generalise a given sample, whereas Section 4 discusses the same question for random vectors.

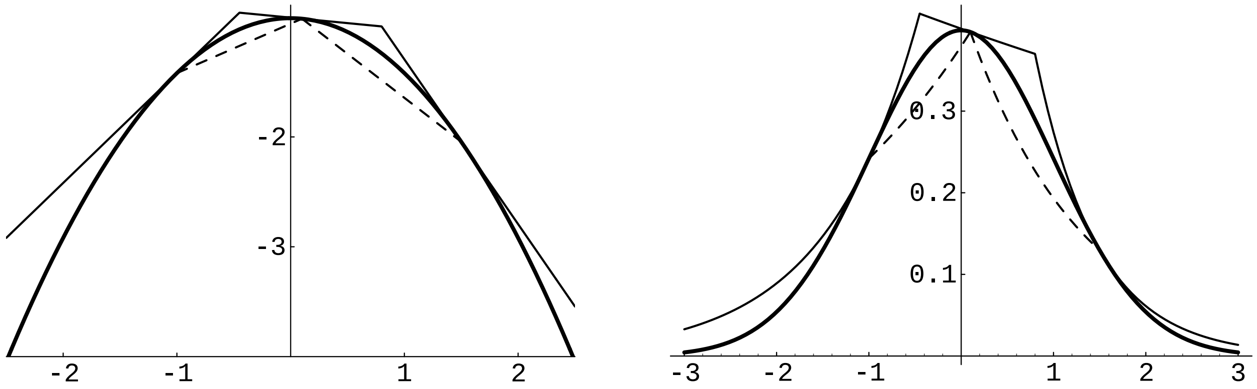

Figure 1: Construction of a hat function for the normal density using transformed density rejection. The left hand side shows the transformed density with three tangents. The right hand side shows the density function with the resulting hat. Squeezes are drawn as dashed lines.

2 UNIVERSAL METHODS

As stated in the introduction there exist several recent universal methods to generate from distributions with known density. Among them a rejection method from a step function (Ahrens 1995) and an adaptive version of the ratio of uniforms method (Leydold 2000). In this paper we want to demonstrate the main ideas of universal algorithms by explaining only the most flexible of the new methods called transformed density rejection.

2.1 Transformed Density Rejection

A pre-version of TDR is already included in Devroye (1986). It was then introduced under a different name by Gilks and Wild (1992), and generalised by Hörmann (1995). It is based on the idea that the given density f is transformed by a strictly monotonically increasing transformation $T\colon (0,\infty) \to \mathbf{R}$ such that $T(f(x))$ is concave. We then say that f is *T-concave*; log-concave densities are an example with $T(x) = \log(x)$.

By the concavity of $T(f(x))$ it is easy to construct a majorising function for the transformed density as the minimum of several tangents. Transforming this function back into the original scale we get a hat function $h(x)$ for the density f. By using secants between the touching points of the tangents of the transformed density we analogously can construct squeezes $s(x)$ (details can be found in Hörmann (1995) or Evans and Swartz (1998). Figure 1 illustrates the situation for the standard normal distribution and $T(x) = \log(x)$. Evans and Swartz (1998) have shown that this technique is even suitable for arbitrary densities provided that the inflection points of the transformed density are known. It should be noted here that the tangent on the transformed density can be replaced by secants through two points that are close together, shifted away from the mode by the distance of these two points. Thus no derivatives are required.

Algorithm TDR applies this idea for a black box algorithm. The I_j are the intervals where the hat h is given by the tangent with touching point c_j.

```
    **Algorithm: TDR**
Require: density $f(x)$; transformation $T(x)$,
    construction points $c_1, \dots, c_n$.
    /* Setup */
1: Construct hat $h(x)$ and squeeze $s(x)$.
2: Compute intervals $I_1, \dots, I_n$.
3: Compute areas $H_j$ below the hat for each $I_j$.
    /* Generator */
4: loop
5:   Generate $I$ with probability vector
       proportional to $(H_1, \dots, H_n)$.
6:   Generate $X$ with density proportional
       to $h|_I$ (by inversion).
7:   Generate $U \sim U(0,1)$.
8:   if $U\,h(X) \le s(X)$ then /* evaluate squeeze */
9:     return $X$.
10:  if $U\,h(X) \le f(X)$ then /* evaluate density */
11:    return $X$.
```

Step 5 is executed in constant time by means of *index search.* Notice that the random variate X is generated by inversion, when random numbers are *recycled* (Devroye 1986, II.3,p.58) and the algorithm is imple-

Distribution	Density	Support	$T_{-1/2}$-concave for
Normal	$e^{-x^2/2}$	$\mathbf{R}$	
Log-normal	$1/x\ \exp(-\ln(x-\mu)^2/(2\sigma^2))$	$[0,\infty)$	$\sigma \leq \sqrt{2}$
Exponential	$\lambda\, e^{-\lambda\, x}$	$[0,\infty)$	$\lambda > 0$
Gamma	$x^{a-1}\, e^{-b\, x}$	$[0,\infty)$	$a \geq 1,\, b > 0$
Beta	$x^{a-1}\,(1-x)^{b-1}$	$[0,1]$	$a, b \geq 1$
Weibull	$x^{a-1}\ \exp(-x^a)$	$[0,\infty)$	$a \geq 1$
Perks	$1/(e^x + e^{-x} + a)$	$\mathbf{R}$	$a \geq -2$
Gen. inv. Gaussian	$x^{a-1}\ \exp(-bx - b^*/x)$	$[0,\infty)$	$a \geq 1,\, b, b^* > 0$
Student's t	$(1+(x^2/a))^{-(a+1)/2}$	$\mathbf{R}$	$a \geq 1$
Pearson VI	$x^{a-1}/(1+x)^{a+b}$	$\mathbf{R}$	$a, b \geq 1$
Cauchy	$1/(1+x^2)$	$\mathbf{R}$	
Planck	$x^a/(e^x - 1)$	$[0,\infty)$	$a \geq 1$
Burr	$x^{a-1}/(1+x^a)^b$	$[0,\infty)$	$a \geq 1,\, b \geq 2$
Snedecor's F	$x^{m/2-1}/(1+m/n\,x)^{(m+n)/2}$	$[0,\infty)$	$m, n \geq 2$

Table 1: $T_{-1/2}$-concave densities (normalisation constants omitted).

mented properly. Therefore we do not recommend the alias method here.

It is obvious that the transformation T must have the property that the area below the hat is finite, and that generating a random variable with density proportional to the hat function by inversion must be easy (and fast). Thus we have to choose the transformations T carefully. Hörmann (1995) suggests the family T_c of transformations, where

$$T_0(x) = \log(x) \quad \text{and} \quad T_c(x) = \operatorname{sign}(c)\, x^c.$$

(sign(c) makes T_c increasing for all c.) For densities with unbounded domain we must have $c \in (-1, 0]$. For the choice of c it is important to note that the area below the hat increases when c decreases. Moreover we find (Hörmann 1995) that if f is T_c-concave, then f is $T_{c'}$-concave for every $c' \leq c$.

Because of computational reasons, the choice of $c = -1/2$ (if possible) is suggested. Then Algorithm TDR can generate random variates of a larger family than the log-concave distributions. Table 1 gives examples of $T_{-1/2}$-concave distributions.

In many simulation books we can find the argument that *inversion* should be used for random variate generation whenever possible, because then correlation induction is easier. But Schmeiser and Kachitvichyanukul (1990) demonstrated how rejection algorithms can be used for correlation induction as well. As the acceptance probability is very high for Algorithm TDR even with moderate n the induced correlation is almost the same as for inversion algorithms (see Hörmann and Derflinger (1994)).

2.2 Construction Points

The above algorithm works well when the ratio between the area below the hat and the area below the squeeze

$$\varrho = \frac{\int h(x)\, dx}{\int s(x)\, dx}$$

is close to one. Thus we have to find construction points, such that ϱ is small.

For the problem of finding appropriate construction points for the hat function Gilks and Wild (1992) have suggested the ingenious concept of *adaptive rejection sampling*. For TDR it works in the following way:

Start with (at least) two points on both sides of the mode and sample points x from the hat distribution. Add a new construction point at x whenever we have to evaluate the p.d.f. $f(x)$, i.e., when $s(x) < U\,h(x)$, until a certain stopping criterion is fulfilled, e.g., the maximal number of construction points or the aimed ratio ϱ is reached.

Obviously the ratio ϱ is a random variable that converges to 1 almost surely when the number n of construction points tends to infinity. A simple consideration gives $\varrho = 1 + O(n^{-2})$ for $c > -1$ (Leydold and Hörmann 1998).

There exist methods for finding construction points such that ϱ is minimised for given number of construction points, transformation and distribution. If only three construction points are used see Hörmann (1995). If more points are required, Derflinger and Hörmann (1998) describe a very efficient procedure.

2.3 Higher Dimensional Distributions

One important feature of the idea of transformed density rejection is that it can be generalised to higher dimensions as well. Again we need a density, a transformation T such that the transformed density is concave and some design points. Then we construct tangential hyperplanes in the design points. The pointwise minimum of these hyperplanes is an upper bound for the transformed density and transformed back by T^{-1} it is thus a hat for the density.

Of course the algorithm becomes more complicated for the higher dimensional case. We have problems to compute the region that corresponds to the interval I_i; we need the sweep-plane technique (Leydold and Hörmann 1998) to generate random variates from the hat; we have sometimes difficulties to find starting points such that the volume below the hat is bounded and we have a lot of numerical difficulties. In Hörmann (2000) all these problems were solved to construct a fast universal algorithm for 2-dimensional log-concave distributions. As optimising the points of contact seems impossible the principle of adaptive rejection sampling is used to get a higher acceptance probability. As original transformed density rejection becomes numerically unstable in higher dimensions, Leydold (1998) presents a variant of transformed density rejection where the $\mathbf{R}^d$ is first decomposed into cones. It works up to dimension 10.

3 GENERALISING A SAMPLE

Assume that we are given a sample $X_1, \ldots, X_n$ of iid random variates. In this case the choice of the input distribution for the stochastic simulation model is a statistical problem, which is called the modeling of probability distributions from data. The problem can be solved in a parametric approach by estimating the parameters of a suitable standard distribution or in a non-parametric approach by estimating the unknown distribution. We are convinced that due to its greater flexibility the non-parametric approach should be used unless there are profound a priori reasons (e.g. of physical nature) favouring a certain standard distribution.

We are interested not only in estimating the input distribution but also in generating random variates from that distribution. This task is called *generating variates from empirical distributions* or *generalising a sample* in the simulation literature. Bratley, Fox and Schrage (1987) and Law and Kelton (1991) have suggested to generate variates by using the linear interpolation of the empirical distribution function. In Hörmann and Bayar (2000) a partly theoretical and partly empirical comparison of the different methods described in the literature show that especially *kernel density estimation* (KDE) is well suited for modeling input distributions for two reasons (Devroye and Györfi (1985), Devroye (1986), Silverman (1986)): Variate generation from these estimates is very simple and the empirical distribution generated is approximating the (unknown) input distribution well as it is based on the mathematical theory of density estimation. To demonstrate the better local approximation of kernel density estimation in a simple example with a triangular distribution see Figure 2.

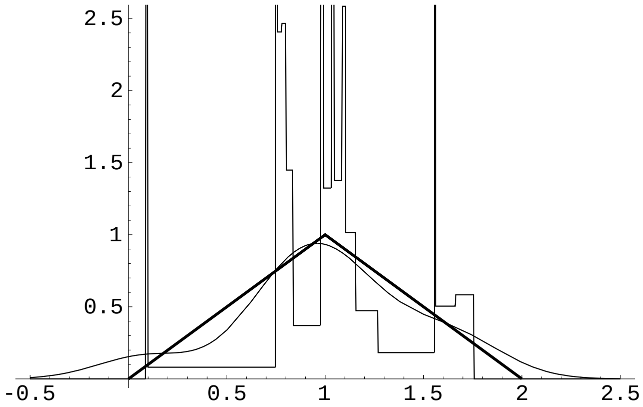

Figure 2: The empirical density of Law and Kelton (1991) (step function) and KDE (smooth function) for a sample of size 20 of the triangular density.

Hörmann and Bayar (2000) conclude that kernel density estimation is superior to the other methods especially with respect to the local behaviour of the distribution. These differences can also influence simulation results for simulations that are sensitive to the fine structure of the stochastic input. We therefore include in this paper only the details of kernel density estimation method.

3.1 Sampling from Kernel Density Estimates

Of course the simplest method of sampling from the empirical distribution is *naive resampling*. We randomly choose members of the sample with replacement. If the sample is based on a continuous random variable this method has the obvious drawback, that only a small number of different values can be generated.

There is a simple modification of naive resampling called *smoothed bootstrap* in the statistic literature. Do not only resample but add some noise to any of the resampled numbers. The noise is a continuous random variable with expectation 0 and small variance. It is not difficult to see, that smoothed bootstrap is the same as generating random variates from a density estimate by using the kernel method, but it is not even necessary to compute the estimated density.

```
Algorithm KDE: (Kernel Density Estimation)
Input: A random sample $X_1, \ldots, X_n$.
Output: A random variate.
0: Setup: Choose the smoothing parameter $b$ (according to formula (1) below).
1: Generate a random integer $I$ uniformly distributed on $\{1, 2, \ldots, n\}$.
2: Generate a random variate $W$ from the noise distribution.
3: Return $Y = X_I + bW$.
```

The density of the random noise distribution W is called *kernel* and will be denoted by $k(x)$. Clearly $k(x)$ must be a density function and it is always assumed to be symmetric around the origin. To change the variance of the random noise we introduce the scale parameter b (called *bandwidth* or smoothing parameter in density estimation); the random variable bW has the density $k(x/b)/b$. The random variate Y generated by Algorithm KDE is the equiprobable mixture of n noise distributions, each centred around one of the sample points. This implies that the density of Y (denoted f_Y) is the sum of n translated versions of $k(x)$ multiplied with $1/n$. f_Y is the kernel density estimate of the unknown distribution and is called $\hat{f}$ in the literature.

$$f_Y(x) = \frac{1}{nb} \sum_{i=1}^{n} k\left(\frac{x - X_i}{b}\right)$$

Of course there remains the question of the choice of the bandwidth b and the kernel function $k(x)$. Here we can use the results of the theory of density estimation as presented e.g. in Silverman (1986) or Wand and Jones (1995). To minimise the mean integrated squared error we use a very simple and robust variant of estimating the optimal bandwidth b as given in Silverman (1986).

$$b = \alpha(k)\, 1.364\, \min(s, R/1.34)\, n^{-1/5} \tag{1}$$

where the constant $\alpha(k)$ is 0.776 for the Gaussian and 1.351 for the rectangular kernel (between -1 and 1) respectively. s denotes the standard deviation and R the interquartile range of the sample. There are lots of much more complicated ways to determine b published in literature. For a survey see Devroye (1997), where the L_1-error (i.e. the mean integrated absolute error) of many different bandwidth selection procedures is compared. The method we use is a mixture of the methods called "reference: L_2, quartile" and "reference: L_2, std. dev" in Devroye (1997). The results of the simulation study show that with the exception of some very strangely shaped multimodal distributions the performance of this very simple choice of b is not bad. And we are not only interested in an optimal estimation of the density here; we also want a good estimate for the CDF and for the moments.

The last question that has to be solved before we can use Algorithm KDE is the choice of the kernel. Asymptotic theory shows that the mean integrated squared error is minimal for the Epanechnikov kernel $f(x) = (1 - x^2)3/4$ but some other kernels have almost the same efficiency. Therefore we can choose the kernel by also considering other properties, e.g. the speed and simplicity of our generation algorithm. In that respect the rectangular kernel (i.e. uniformly distributed noise between -1 and 1) is of course the best choice, but it has the theoretical drawback that the estimated density is not continuous. Due to the nice statistical interpretation Gaussian noise is a good choice as well.

Algorithm KDE guarantees that the density function of the empirical distribution approximates the density of the unknown true distribution as good as possible with respect to the mean integrated squared error. On the other hand it is not difficult to show, that for algorithm KDE the variance of the empirical distribution is always larger than the variance of the observed sample. This can be a disadvantage in simulations that are sensitive against changes of the variance of the input distributions. To overcome this problem it is possible to force the empirical distribution to have the same variance as the sample in the following way (Silverman 1986).

```
Algorithm KDEVC: KDE variance corrected
Input: A random sample $X_1, \ldots, X_n$.
Output: A random variate.
0: setup: Compute the mean $\bar{x}$, the standard deviation $s$ and the interquartile range $R$ of the sample. Compute $b$ according to formula (1) and $c_b = 1/\sqrt{1 + b^2\sigma_k^2/s^2}$ ($\sigma_k^2$ denotes the variance of the kernel).
1: Generate a random integer $I$ uniformly distributed on $\{1, 2, \ldots, n\}$.
2: Generate a random variate $W$ from the noise distribution.
3: Return $Y = \bar{x} + (X_I - \bar{x} + bW)c_b$.
```

Remark: Positive random variables are interesting for many applications. Method KDE can cause problems for such applications as it will also generate negative variates. The easiest way out is the so-called mirroring principle. Instead of a negative number Y simply return $-Y$. Unfortunately the mirroring principle disturbs the variance correction. They can be used together but the resulting empirical distribution has a smaller variance than the sample. This can only be a practical problem if the sample of a positive distribution has many values close to zero.

Remark: From the correlation induction perspective method KDE has the disadvantage that it is no inversion method. One can try to induce some correlation by ordering the sample but the result will be satisfactory only for very large samples. Another possibility is to generate from the kernel density estimate by inversion. If we use a rectangular kernel the resulting algorithm is not too complicated. We get a slower setup step and higher memory requirements but the good approximation properties and the fast marginal generation time remain. Unfortunately this inversion approach spoils the simplicity and elegance of Algorithm KDE but it is not much more complicated than a linear interpolation of the empirical distribution function.

4 GENERALISING A VECTOR-SAMPLE

When modeling the stochastic input of a simulation model we often have the problem of dependencies between different input values. One of the most common modeling errors is to overlook these dependencies. Nelson and Yamnitsky (1998) describe a method where it is only necessary to specify the marginal distributions and the correlation structure of the random vector which may be attractive for many users. But the only possibility to fully model dependencies between input variables is to specify the multidimensional distribution of that random vector. Devroye (1986) and Johnson (1987) describe the generation of random vectors for many distributions including the multivariate Johnson family. Wagner and Wilson (1995) describe the bivariate Bezier distribution. Another possibility is to specify a log-concave density function and to use multidimensional transformed density rejection (as shortly described in Section 2.3). If enough data are available we can also use a method to generalise a sample of vectors. This can be of great practical importance as there are very few methods that can be easily used to model and generate random vectors from data.

Possible algorithms include using the *multi-normal distribution* that has the same expectation and covariance matrix as the given sample, *naive resampling* and a method described by Taylor and Thompson (1986) that samples from a mixture of nearest neighbour and kernel density estimate, which is also included in the IMSL library. As it is always the case the difficulty of estimation increases with the dimension and we need really large samples to obtain acceptable estimates of the unknown density.

4.1 Sampling from Multidimensional Kernel Density Estimates

The idea (resampling plus noise) remains the same as in dimension one and we can leave Algorithm KDE unchanged and use it as vector algorithm by just interpreting X_I and W as vectors. But we have to specify the full covariance matrix of the noise. This problem is more difficult than in dimension one and there is no generally accepted method recommended in the literature. Silverman (1986) explains a "quick and dirty" version that is based on the following idea: We can transform the data of the sample such that they have unit covariance matrix. Then we can use (as for dimension one) a simple reference method to find the smoothing parameter b of a radial symmetric kernel. For the normal kernel the formula is:

$$b = \left(\frac{4}{(d+2)n}\right)^{\frac{1}{d+4}} \tag{2}$$

where n denotes the sample size and d the dimension of the vectors. After adding the noise to the chosen transformed data point we transform them back to the original covariance structure. It is not difficult to see that instead of transforming the data we can equivalently use a kernel, that has the same covariance matrix as the data and is multiplied by smoothing parameter b. As in dimension one the above b is oversmoothing in many cases where the unknown distribution is multimodal or skewed. Therefore it is of course possible to use a smaller b (for example half of it) if we assume that the unknown distribution is strongly non-normal. We know that the estimate of b can be far from optimal for many distributions. Therefore we suggest to use the normal kernel and the variance corrected version here. We can show that KDEVCvec contains fitting the normal distribution and naive resampling as limiting cases for bandwidth b towards infinity and bandwidth towards 0 respectively. Fitting the multi-normal distribution ($b \to \infty$) is best if the (unknown) distribution is normal, naive resampling ($b = 0$) is optimal if the unknown distribution is discrete. In other cases it seems obvious that values of b in between lead to a better approximation of the unknown distribution. So even if our guess of b is far from optimal it is still very likely that it is better than using $b = 0$ or $b = \infty$ which shows that Algorithm KDEVCvec should have better properties than naive resampling or fitting the normal distribution for most continuous distributions. Collecting the details we can state:

```
**Algorithm KDEVCvec**:
Input: A random sample $X_1, \ldots, X_n$ of vectors.
Output: A random vector of length $d$.
0: setup: Compute the mean vector $\bar{x}$, the covariance matrix $s$ and the Cholesky-factor $l$ of $s$.
   Compute $b$ using formula (2) and $c_b = \sqrt{1+b^2}$.
1: Generate a random integer $I$ uniformly distributed on $\{1, 2, \ldots, n\}$.
2: Generate a random vector $W$ of $d$ independent normal variates.
3: Return $Y = \bar{x} + (X_I - \bar{x} + l(bW))c_b$.
```

Compared to the method of Taylor and Thompson (1986) KDEVCvec has the advantage that it is based on a well understood method of density estimation whereas no theory is available to show the approximation properties of the Taylor and Thompson (1986) method. Another practical disadvantage of that method is that the sampling is much slower than for the kernel method and that the setup time and the storing requirements explode for large samples and higher dimensions.

5 CONCLUSIONS

We have introduced two approaches for non-uniform random variate generation for simulation. The first one uses transformed density rejection and is based on the knowledge of the density of the input distribution. The second approach requires a (not too small) random sample from the unknown distribution and – implicitly modeling the distribution by using kernel density estimation – directly generates random variates.

For continuous distributions these two approaches can solve most random variate generation problems that occur in simulation studies. One of their main advantages is the fact that both methods work also for random vectors which means that they can solve problems that are not included in random variate generation libraries available up to now.

Therefore the algorithms described in this paper (and many others) have been implemented in C by the authors in a library called UNURAN which is available via anonymous ftp (Leydold and Hörmann 2000).

ACKNOWLEDGEMENTS

This work was supported by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung*, Proj. 12805-MAT and by the Austrian Academy of Science, APART-scholarship.

REFERENCES

Ahrens, J., H. 1995. A one-table method for sampling from continuous and discrete distributions. *Computing*, 54(2):127–146.

Bratley, P., B. L. Fox, and L. E. Schrage. 1987. *A guide to simulation.* 2d ed. New York: Springer-Verlag.

Dagpunar, J. 1988. *Principles of Random Variate Generation.* Oxford, U.K.: Clarendon Oxford Science Publications.

Derflinger, G. and W. Hörmann. 1998. The optimal selection of hat functions for rejection algorithms. Unpublished manuscript.

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

Devroye, L. 1997. Universal smoothing factor selection in density estimation, theory and practice. *Test*, 6:223–320.

Devroye, L. and L. Györfi. 1985. *Nonparametric Density Estimation: The L_1 View.* New-York: John Wiley.

Evans, M. and T. Swartz. 1998. Random variable generation using concavity properties of transformed densities. *Journal of Computational and Graphical Statistics*, 7(4):514–528.

Gentle, J. E. 1998. *Random Number Generation and Monte Carlo Methods.* Statistics and Computing. New York:Springer.

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

Hörmann, W.1995. A rejection technique for sampling from T-concave distributions. *ACM Transactions on Mathematical Software*, 21(2):182–193.

Hörmann, W. 2000. An automatic generator for bivariate log-concave distributions. *ACM Transactions on Modeling and Computer Simulation*, (to appear).

Hörmann, W. and O. Bayar. 2000. Modeling probability distributions from data and its influence on simulation. In Troch I. and Breitenecker F., editors, *Proceedings IMACS Symposium on Mathematical Modeling*, Argesim Report No. 15, pages 429 – 435.

Hörmann, W. and G. Derflinger. 1994. Universal generators for correlation induction. In *Compstat, Proceedings in Computational Statistics*, ed. R. Dutter and W. Grossmann, 52–57. Physica-Verlag, Heidelberg.

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

Law, A. and D. Kelton. 1991. *Simulation Modeling and Analysis.* New York: Mc-Graw-Hill.

Leydold, J. 1998. A rejection technique for sampling from log-concave multivariate distributions. *ACM Transactions on Modeling and Computer Simulation*, 8(3):254–280.

Leydold, J. 2000. Automatic sampling with the ratio-of-uniforms method. *ACM Transactions on Mathematical Software*, (to appear).

Leydold, J. and W. Hörmann. 1998. A sweep-plane algorithm for generating random tuples in simple polytopes. *Mathematics of Computation*, 67(224):1617–1635.

Leydold, J. and W. Hörmann. 2000. *UNURAN – A library for Universal Non-Uniform Random number generators.* Institut für Statistik, WU Wien, A-1090 Wien, Austria. available at ftp://statistik.wu-wien.ac.at/src/unuran/.

Nelson, B. L. and M. Yamnitsky. 1998. Input modeling tools for complex problems. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, 105–112. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schmeiser, B. W. and V. Kachitvichyanukul. 1990. Non-inverse correlation induction: guidelines for algorithm development. *Journal of Computational and Applied Mathematics*, 31: 173–180.

Silverman, B. 1986. *Density Estimation for Statistics and Data Analysis.* London: Chapman and Hall.

Taylor, M.S. and J. R. Thompson. 1986. A data based algorithm for the generation of random vectors. *Comput. Statist. Data Anal.*, 4:93–101.

Wagner, M. A. F. and J. R. Wilson. 1995. Graphical interactive simulation input modeling with bivariate Bezier distributions. *ACM TOMACS*, 5:163–189.

Wagner, M. A. F. and J. R. Wilson. 1996. Using univariate Bezier distributions to model simulation input processes. *IIE Transactions*, 28:699–711.

Wand, M. and M. Jones. 1995. *Kernel Smoothing.* London: Chapman and Hall.

AUTHOR BIOGRAPHIES

WOLFGANG HÖRMANN is visiting researcher in the Department of Industrial Engineering at Boğaziçi University Istanbul. He received M.S. and Ph.D. degrees in mathematics from the University of Vienna, and was associate professor in the Department of Statistics at the University of Economics and Business-Administration Vienna. His email adress is <hormannw@boun.edu.tr>.

JOSEF LEYDOLD is an assistant professor in the Department of Statistics at the University of Economics and Business-Administration Vienna. He received M.S. and Ph.D. degrees in mathematics from the University of Vienna. His email address is <Josef.Leydold@statistik.wu-wien.ac.at>.