Inverse Transformed Density Rejection for Unbounded Monotone Densities

Hörmann, Wolfgang; Leydold, Josef; Derflinger, Gerhard

DOI:
10.1145/1276927.1276931

Published: 01/01/2007

Document Version
Publisher's PDF, also known as Version of record

Automatic Random Variate Generation for Unbounded Densities

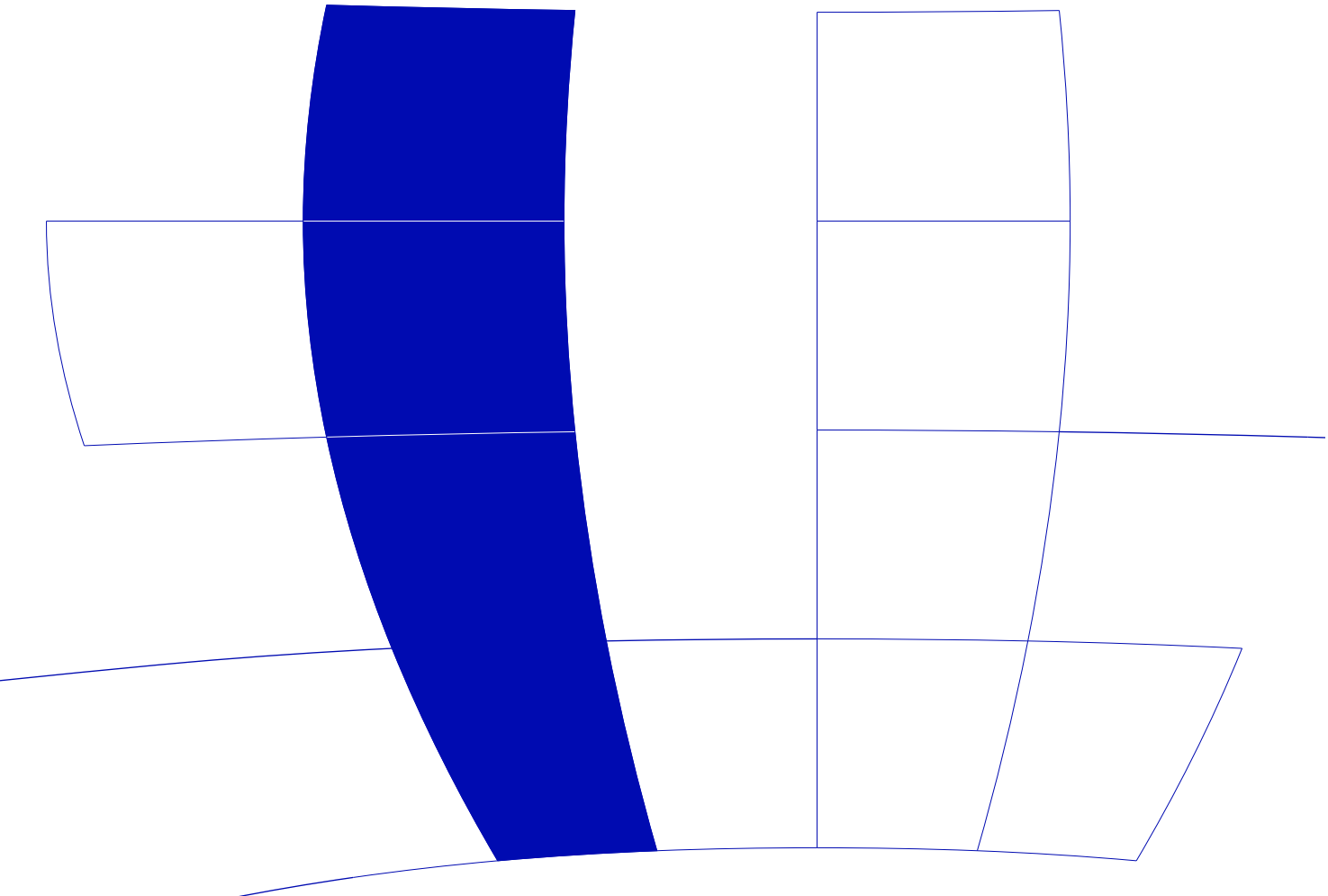
Wolfgang Hörmann, Josef Leydold, Gerhard Derflinger

Department of Statistics and Mathematics
Wirtschaftsuniversität Wien

Research Report Series

Report 57
August 2007

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



Inverse Transformed Density Rejection for Unbounded Monotone Densities

WOLFGANG HÖRMANN, JOSEF LEYDOLD, AND GERHARD DERFLINGER
Department of Statistics and Mathematics
Vienna University of Economics and Business Administration

A new algorithm for sampling from largely arbitrary monotone, unbounded densities is presented. The user has to provide a program to evaluate the density and its derivative and the location of the pole. Then the setup of the new algorithm constructs different hat functions for the pole region and for the tail region, respectively. For the pole region a new method is developed that uses a transformed density rejection hat function of the inverse density. As the order of the pole is calculated in the setup, conditions that guarantee the correctness of the constructed hat functions are provided. Numerical experiments indicate that the new algorithm works correctly and moderately fast for many different unbounded densities.

Categories and Subject Descriptors: G.3 [**Probability and Statistics**]: Random number generation

General Terms: Algorithms

Additional Key Words and Phrases: non-uniform random variates, universal method, black-box algorithm, transformed density rejection, unbounded densities

1. INTRODUCTION

Automatic algorithms (also called universal or black-box algorithms) are an important development in random variate generation (see the recent survey by Hörmann et al. [2004] and the references given there). Such algorithms are applicable to (often large) families of densities. The user typically has to provide a function that evaluates the density of the target distribution and sometimes some extra information like the (approximate) location of the mode. In a setup step the automatic algorithm calculates all constants necessary to run the sampling part of the algorithm which then generates variates from the desired distribution. The obvious advantage of automatic algorithms is their flexibility. A single algorithm coded, tested and investigated only once can be used to sample from many different distributions. Of course we cannot expect that an automatic algorithm works for all distributions. Therefore, in applying an automatic algorithm the user has to know if the algorithm is able to construct a correct generator for the distribution she is interested in. Normally this is done by checking that the distribution fulfills certain

Author's address: Vienna University of Economics and Business Administration, Department of Statistics and Mathematics, Augasse 2-6, A-1090 Vienna, Austria
email: leydold@statistik.wu-wien.ac.at
This work was supported by the Austrian Science Foundation (FWF), project no. P16767-N12.

conditions. For example, in the literature there exist several automatic algorithms for log-concave distributions. Before applying them to a distribution we should check whether the density is log-concave which is often no problem if the density is available in closed form. However, it is also possible that the algorithm itself checks its correctness at generation time. This is convenient for the user if the density of the distribution is not available in closed form or checking the conditions is cumbersome. It is a matter of taste if such an approach should still be called *automatic*. We are of that opinion and we are convinced that this approach of checking the algorithm at generation time is attractive for many users.

Transformed density rejection (TDR) is an example of an automatic algorithm. It allows to generate from a large class of bounded, unimodal densities [Hörmann 1995]. This class contains important standard distributions like Gamma, Beta and t-distributions together with less well known distributions like Planck, Hyperbolic and the generalized inverse Gaussian distributions. It is also possible to apply this (and similar automatic) methods to sample from distributions with multimodal densities by decomposing the domain to get a composition of unimodal distributions. But there are no (exact) universal algorithms available in the literature that are applicable to unbounded densities. Nevertheless, sampling from such distributions is of practical relevance, because many of the best known standard distributions have parameter regions where their densities contain poles. Note that Markov Chain Monte Carlo algorithms could be used but they do not produce iid. samples.

In this paper we introduce an extension of transformed density rejection applicable to monotone unbounded densities. When designing rejection algorithms for monotone densities it is useful to observe that any monotone density can always be cut into a density on $(0, 1)$ with a (possible) pole at 0 and a bounded density on $(1, \infty)$. There is literature on how to deal with the tail part. So the fundamental design problem is how to deal with a monotone density on $(0, 1)$. By permuting x and y it is possible to transform this design problem into that of generating from a bounded density on $(1, \infty)$. For obtaining a good algorithm it turns out that the actual way of permuting x and y and even more the selection of the point where the two regions are cut are of main importance as we will see in Section 3.

The paper is organized as follows: Section 2 explains the idea of TDR and states the results of the literature that are used to develop our algorithm. (For further details we refer the interested reader to Hörmann et al. [2004, Chap. 4].) Section 3 introduces the new idea of *inverse transformed density rejection* (ITDR) and uses it to develop an automatic algorithm for monotone unbounded densities. We present some theorems on correctness and theoretical performance of the new method. Section 4 discusses the implementation and the numerical stability of the new algorithm.

An implementation of this algorithm is available with the UNU.RAN library for Universal Non-Uniform RANdom variate generation [Leydold and Hörmann 2006].

2. TRANSFORMED DENSITY REJECTION

The basic idea of *transformed density rejection* (TDR) is simple: The given density f is transformed by a strictly monotonically increasing transformation $T\colon (0, \infty) \to \mathbb{R}$ 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 exploiting the concavity of $T(f(x))$ it is easy to construct an upper bound for the transformed density as the minimum of one, two or more tangents. Transforming this function back into the original scale, we get a hat function $h(x)$ for the density f. Each of the tangents defines an interval where the "TDR hat" is given by $h(x) = T^{-1}(\alpha(p) + \beta(p)\, x)$ where p denotes the point of contact of the tangent (also called *design point*), $\beta(p) = T'(f(p))\, f'(p)$ and $\alpha(p) = T(f(p)) - p\,\beta(p)$. For an interval (b_0, ∞) with design point p a straightforward computation yields for the area below the hat

$$A_h = -F_T(\alpha(p) + \beta(p)\, b_0)/\beta(p) \tag{1}$$

where F_T denotes an anti-derivative of the inverse transformation T^{-1}, see [Hörmann et al. 2004, p. 59]. Notice that $\beta(p)$ must be less than zero since otherwise the area is not finite. With an abuse of language, we call the integral $H(x) = \int_{b_0}^{x} h(t)\, dt$ the cumulative distribution function (CDF) of the hat function. For its inverse we find

$$H^{-1}(u) = [F_T^{-1}(\beta(p)\, u + F_T(\alpha(p) + \beta(p)\, b_0)) - \alpha(p)]/\beta(p) \quad \text{for } 0 \le u \le A_h. \tag{2}$$

It is obvious that the transformation T must have the property that the area A_h below the hat h 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 increasing transformations that are similar to the well-known Box-Cox transformations commonly used in statistics [Box and Cox 1964]. We define

$$T_0(x) = \log(x) \quad \text{and} \quad T_c(x) = -\,x^c \text{ for } c < 0\,. \tag{3}$$

Notice that the notion of T_c-concave distributions is more general than log-concave distributions and includes also unimodal densities with heavy tails.

For the TDR algorithms of this paper we need the inverse transform $T^{-1}(x)$, its antiderivative $F_T(x)$ and the inverse of $F_T(x)$. All of these are defined for $c < 0$, $c \neq 0$, by:

$$T_c^{-1}(x) = (-\,x)^{1/c}\,, \quad F_{T_c}(x) = \frac{-(-x)^{\frac{c+1}{c}}}{\frac{c+1}{c}}\,, \quad F_{T_c}^{-1}(x) = -\left(-x\,\frac{c+1}{c}\right)^{\frac{c}{c+1}}\,. \tag{4}$$

For the major steps in the design of an algorithm for a particular distribution based on TDR one has to select a proper value for the parameter c as well as the number and location of the design points. Thus the following has to be taken into consideration:

—For densities with unbounded domain we must have $c > -1$ as otherwise the area below the hat is not bounded.

—For fixed design points the area A_h below the density (and thus the rejection constant) increases when c decreases.

—If f is T_c-concave, then f is $T_{c'}$-concave for every $c' \le c$.

As a consequence one should choose c as large as possible. For densities with heavy tails values of c close to -1 must be used. For finding appropriate values for c the

notion of *local concavity* as introduced by Hörmann et al. [2004, p. 66] is useful. It is the maximum value of c such that $T_c(f(x))$ is locally concave in x. For differentiable densities it is given by

$$\mathrm{lc}_f(x) = 1 - \frac{f''(x)\, f(x)}{f'(x)^2}\,. \tag{5}$$

The maximal possible c to reach T_c-concavity is then the infimum of $\mathrm{lc}_f(x)$ over the domain of the given density.

For a particular choice of c the rejection constant depends on the number and the location of the design points. Many design points result in fast algorithms with slower setup and higher memory consumption whereas three or less design points lead to simpler algorithms with shorter setup, see Hörmann et al. [2004] for different variants of TDR and the selection of the design points. In this paper we are only interested in simple algorithms for monotone distributions on $(0, \infty)$. Thus we make use of the following two results on the optimal choice of one and two design points, respectively.

THEOREM 1 [HÖRMANN AND DERFLINGER 1996; HÖRMANN ET AL. 2004]. *Let $f(x)$ be a monotone strictly T_c-concave density on (b_0, ∞). When a hat function is constructed by means of a single tangent then the optimal single design point p has the property*

$$(p - b_0)f(p) = \max_x (x - b_0)f(x) \quad \textit{or equivalently} \quad (p - b_0)\, f'(p) + f(p) = 0\,.$$

The optimal design point does not depend on c. Denoting the area below the density by A_f, the rejection constant of the optimal hat with single design point p_0 is bounded by:

$$\alpha = \frac{A_h}{A_f} \le (1 + c)^{1/c} \textit{ for } -1 < c < 0 \qquad \textit{and} \qquad \frac{A_h}{A_f} \le \mathrm{e} \textit{ for } c = 0.$$

THEOREM 2 [HÖRMANN 1995; HÖRMANN ET AL. 2004]. *Let f be a monotone T_c-concave density on $(0, \infty)$. Assume that a hat function is constructed by means of a horizontal tangent in mode $p_m = 0$ and a second tangent in design point p_t. Then the area below the hat is minimized when p_t fulfills the condition*

$$f(p_t) = f(0)\,(1 + c)^{-1/c} \;\textit{ for } c < 0 \qquad \textit{and} \qquad f(p_t) = f(0)/e \;\textit{ for } c = 0.$$

The rejection constant of the optimal hat with three design points is bounded by:

$$\alpha = \frac{A_h}{A_f} \le \frac{1}{1 - 1/(1 + c)^{1+1/c}} \;\textit{ for } -1 < c < 0 \quad \textit{and} \quad \frac{A_h}{A_f} \le \frac{\mathrm{e}}{\mathrm{e} - 1} \;\textit{ for } c = 0.$$

2.1 Transformed Density Rejection and Poles

Assume, for example, the density of the Gamma distribution with shape parameter $a < 1$. Its local concavity is given by $(a - 1)/(a - 1 - x)^2$ and thus it is T_c-concave only for $c \le 1/(a - 1) < -1$ which leads to a hat with unbounded integral. It is not difficult to show that for every density with pole there are points where the local concavity is smaller than or equal to -1.

THEOREM 3. *A density with pole cannot be T_c-concave for any $c > -1$.*

PROOF. Let f be a T_c-concave density with domain $(0, b)$ and a pole at 0. Thus $\lim_{x\to 0} T_c(f(x)) = 0$ and $x\,T_c(f(b))/b$ is a lower bound for the transformed density. Hence

$$s(x) = T_c^{-1}(x\,T_c(f(b))/b) = x^{\frac{1}{c}}\,f(b)/(b^{\frac{1}{c}})$$

is a lower bound for f. Consequently, $c < -1$ since otherwise $\int_0^b s(x)\,dx \leq \int_0^b f(x)\,dx$ diverges. □

The above theorem implies that a TDR algorithm with one single parameter c is not applicable to a density with pole and domain $(0, \infty)$. But it is possible to use TDR with different values of c in at least two intervals. We tried the Gamma, Beta, Beta prime, F and Planck distributions and noticed that for all of them $\mathrm{lc}_f(x)$ is monotonically increasing. The limit $\lim_{x\to 0} \mathrm{lc}_f(x)$ is approaching -1 for the pole becoming heavier and $-\infty$ for the pole becoming very light. Therefore we can define $c_p = \lim_{x\to 0} \mathrm{lc}_f(x)$ and know that f is c_p concave due to the monotonicity of the local concavity lc_f. Nevertheless c_p can only be used on a bounded interval $(0, b_r)$ with b_r finite as otherwise the hat function has unbounded integral. For the remaining part (b_r, ∞) a different value $c_t > -1$ for the transformation T_c has to be chosen. Moreover, one of the design points must be the pole itself. This is the approach suggested by Evans and Swartz [1998] for the F-distribution when its density is unbounded.

When we want to apply this idea for creating a simple algorithm with only one design point in the pole region $(0, b_r)$ we have to use 0 (the pole). Then the transformed hat can be written as $T_{c_p}(h(x)) = k\,x$ where k is $\lim_{x\to 0} T(f(x))'$ and the hat function is therefore a multiple of x^{1/c_p}. Thus the direct TDR approach in the interval containing the pole leads to the hat function that is used in many rejection algorithms suggested in the literature for densities with a polynomial pole. However, this hat is not optimal as it is touching the density in the pole itself.

We used this direct TDR approach for the distributions mentioned above to construct a TDR algorithm with just two intervals, using $c = -0.5$ for the right interval. The resulting algorithm has a rejection constant above 1.4 for the case that the pole is very light which is certainly not satisfactory. We therefore develop a new method that allows to construct a larger variety of TDR-like hats to a density with pole. One advantage of the new approach is that those hats do not all touch in 0.

It is well known that one can make a pole at the origin disappear by applying the Box-Cox transformation to the variate x itself, i.e., by means of the transformation $x \mapsto x^\gamma$ for some $\gamma > 1$. The algorithm of Cheng and Feast [1980] for the Gamma distribution is such an example for this alternate approach. However, it deserves more studies to find simple algorithms that allow the choice of parameter γ as well as the construction of the hat function automatically.

3. INVERSE TRANSFORMED DENSITY REJECTION

3.1 The Main Idea

In the following we assume that f is a monotone differentiable density on $(0, \infty)$ with a pole at $x = 0$. For the design of our proposed algorithm we decompose

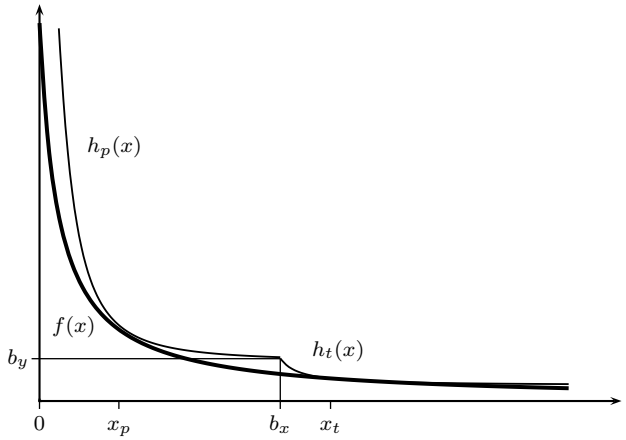


Fig. 1. Hat function for inverse transformed density rejection (ITDR)

the density into two parts: The tail region with $x > b_x$ and the pole region with $x \le b_x$, see Figure 1. For the tail region a TDR algorithm with a single design point x_t and concavity parameter c_t is applied. For the pole region we consider the inverse density $f^{-1}(y)$ instead of the density itself. Thus the pole region of the density is transformed into the density $\min(f^{-1}(y), b_x)$ with unbounded tail that can be easily handled by TDR algorithms. Here a reader may get the impression that there is little new in that straightforward idea and that the generation is the same for the tail and for the pole region. This is correct but there are at least three important problems we have to solve to obtain a competitive algorithm. First, we have to formulate the new variant of transformed density rejection such that the inverse density is never evaluated. We call the new approach *Inverse Transformed Density Rejection* (ITDR) where we use the mode at $y = 0$ and one additional point as design points for the hat of this part of the density. Secondly, we have to select the point b_x, the border between the pole region and the tail region. The third problem is related with the fact that even after decomposing the density the tail part is often only T_c-concave for a value of c only slightly larger than -1. In that case Theorem 1 implies that standard TDR with a single optimal design point leads to quite large rejection constants α. To overcome this problem we develop a new version of TDR with a single design point that allows the use of larger values of c and thus leads to better fitting hats in the tail region.

Addressing the details we construct the hat function h_p on the pole region $(0, b_x)$ by fitting a function to the inverse density. In other words the hat function has the inverse function

$$h_p^{-1}(y) = T_c^{-1}(\alpha(x_p) + \beta(x_p)\, y)\,,$$

where the constants $\alpha(x_p)$ and $\beta(x_p)$ have to be chosen such that $h_p(x_p) = f(x_p)$ and $h_p'(x_p) = f'(x_p)$ for the design point x_p of the tangent. By inverting the above definition we find

$$h_p(x) = \frac{T_c(x) - \alpha(x_p)}{\beta(x_p)} \qquad \text{and} \qquad h_p'(x) = \frac{T_c'(x)}{\beta(x_p)}$$

and consequently we get

$$\beta(x_p) = \frac{T_c'(x_p)}{f'(x_p)} \qquad \text{and} \qquad \alpha(x_p) = T_c(x_p) - f(x_p)\,\beta(x_p)\ .$$

Inverting the cumulative distribution function (CDF) of the hat function $h_p(x)$ in order to generate X is computationally too expensive. It is much easier to generate the y-coordinate of a random point (X, Y) first by inverting the CDF of the inverse $h_p^{-1}(y)$ of the hat. The x-coordinate is uniformly distributed on $(0, h_p^{-1}(Y))$ and we check the condition $Y \le f(X)$ to decide whether we can accept X. Notice that it is not necessary to evaluate the inverse of the density.

The inverse hat $h_p^{-1}(y)$ is constant for $y \in (0, b_y)$ where $b_y = h_p(b_x)$. In this case generating Y is simple. For the interval (b_y, ∞) the area A_p and the inverse $H_p^{-1}(u)$ of the CDF of the inverse hat function are given by (1) and (2) as

$$A_p = -F_T(\alpha(x_p) + \beta(x_p)\,b_y)/\beta(x_p) \tag{6}$$

and

$$H_p^{-1}(u) = [F_T^{-1}(\beta(x_p)\,u + F_T(\alpha(p) + \beta(p)\,b_y)) - \alpha(p)]/\beta(p) \quad \text{for } 0 \le u \le A_p. \tag{7}$$

The main idea of the algorithm can now be easily formulated:

Setup:

1. Select a point b_x such that $\mathrm{lc}_f(x) > -1$ for all $x > b_x$ and $\mathrm{lc}_{f^{-1}}(f(x)) > -1$ for all $x \le b_x$.
2. Find c_t for the tail region and c_p for the pole region.
3. Select design points x_t and x_p for the tail region and the pole region, respectively.
4. Compute respective areas A_t, A_c and A_p for the tail region, the *central* rectangle $(0, b_x) \times (0, b_y)$, and the upper pole region with $y > b_y$.

Generator:

5. Choose one of the tail, central and upper pole region at random with probability vector proportional to (A_t, A_c, A_p).
6. Generate a point (X, Y) uniformly in the chosen region.
7. If $Y \le f(X)$ return X.
8. Otherwise, try again.

The algorithm is thus fairly simple. The main problem left is the selection of b_x, a good choice of the TDR parameters c_t and c_p and of the design points x_t and x_p. A possible approach is to use tools like `Matlab` or `Mathematica` to plot and analyze $\mathrm{lc}_f(x)$ and $\mathrm{lc}_{f^{-1}}(y)$, select b_x such that both c_t and c_p are as large as possible and use Thms. 1 and 2 to find proper design points for the respective hats. In the following we suggest simple rules for the parameter selection and proof that they are guaranteed to work properly if the density f fulfills a fairly mild condition.

3.2 Selecting b_x

To be able to apply the above idea of ITDR to unbounded monotone densities we have to decide on the border b_x between pole region and tail region. For this task

it is convenient to express the local concavity of the inverse density at some point y as a function of $x = f^{-1}(y)$. Using the formulas for the first and second derivative of the inverse function, $(f^{-1})'(y) = 1/f'(x)$ and $(f^{-1})''(y) = -f''(x)/f'(x)^3$, we arrive at

$$\mathrm{ilc}_f(x) = 1 + x\,\frac{f''(x)}{f'(x)} \tag{8}$$

which we call the *inverse local concavity* of f at x. Notice that f must satisfy $\mathrm{ilc}_f(x) > -1$ in a (at least) small environment of the pole in order to apply ITDR.

Theorem 3 and the necessary condition that the inverse density must be T_c-concave for a $c > -1$ near the pole implies that we have $\mathrm{lc}_f(x) < \mathrm{ilc}_f(x)$ for x close to 0. As the inverse local concavity ilc_f of the density f is defined as the local concavity $\mathrm{lc}_{f^{-1}}(f(x))$ of the inverse density $f^{-1}(y)$ and vice versa it follows that $\mathrm{lc}_f(x) < \mathrm{ilc}_f(x)$ for large x. Therefore $\mathrm{lc}_f(x)$ and $\mathrm{ilc}_f(x)$ have an intersection point. Experimenting with many unbounded densities we have observed that most densities of interest have a single intersection point of $\mathrm{lc}_f(x)$ and $\mathrm{ilc}_f(x)$. This point x_i can be found easily by a search algorithm as the equation $\mathrm{lc}_f(x_i) = \mathrm{ilc}_f(x_i)$ simplifies to $x_i\,f'(x_i) + f(x_i) = 0$. This means that x_i is also the point leading to the largest rectangle $(0, x) \times (0, f(x))$ below the density. The following lemma verifies our observations.

LEMMA 4. *For a twice differentiable unbounded density f with domain $(0, \infty)$ there exists a local maximum x_i of $xf(x)$ which is the smallest x_i with $\mathrm{lc}_f(x_i) = \mathrm{ilc}_f(x_i)$.*

PROOF. Suppose that no point with $\mathrm{lc}_f(x_i) = \mathrm{ilc}_f(x_i)$ exists. Then $x\,f'(x) + f(x) = (x\,f(x))'$ is either always greater than 0 or less than 0. In the first case $x\,f(x)$ is monotonically increasing and we find $f(x) \geq f(1)/x$ for all $x \geq 1$. Hence f cannot have a bounded integral, a contradiction. An analogous contradiction follows in the second case where $f(x) \geq f(1)/x$ for all $x \leq 1$ as $x\,f(x)$ is decreasing. Hence the smallest point with $x\,f'(x) + f(x) = 0$ must be a (local) maximum of $x\,f(x)$. □

Note that it is possible to construct examples of convex monotone densities where lc_f and ilc_f have more than one intersection point. For example, the density $f(x) = x^{-0.9}(1 + \cos(10\,\pi\,x)/100)$ is monotone and convex on $(0, 0.55)$, has a $T_{-0.9}$-concave mode and more than one intersection point. In such cases we define x_i as the leftmost intersection point.

Due to Lemma 4 the situation that an intersection point does not exist is only possible if b_r, the right border of the domain, is finite. In such a case we can use $x_i = b_r$ and the algorithm only generates from the pole region.

It is quite clear that our simple general approach for ITDR may not work properly if $\mathrm{ilc}_f(x_i) = \mathrm{lc}_f(x_i) \leq -1$. The following result shows that only equality can occur.

THEOREM 5. *Let $f(x)$ be a twice continuously differentiable strictly monotone density with domain $(0, \infty)$ and let x_i be the intersection point of Lemma 4. Then $\mathrm{lc}_f(x_i) = \mathrm{ilc}_f(x_i) \geq -1$. Equality holds if and only if $x_i^2\,f''(x_i) = 2\,f(x_i)$.*

PROOF. By Lemma 4, x_i is a local maximum of $xf(x)$. Then $f(x) \leq g(x) = x_i f(x_i)/x$ for all x in a sufficiently small interval $(x_i - \varepsilon, x_i + \varepsilon)$. Clearly $f(x_i) =$

$g(x_i)$ and it is easy to show that $f'(x_i) = g'(x_i) < 0$ and $f''(x_i) \leq g''(x_i)$ as $f(x) - g(x)$ has a local maximum at x_i. Consequently, $\mathrm{lc}_f(x_i) = \mathrm{ilc}_f(x_i) = 1 + x_i f''(x_i)/f'(x_i) \geq 1 + x_i g''(x_i)/g'(x_i) = -1$. Equality holds if and only if $f''(x_i) = g''(x_i)$. □

A natural approach for the task of finding b_x is to use this intersection point x_i as the first candidate. However, in our experiments it turned out that in the case where c_p (see Section 3.3.1 for a selection rule) is close to -1 this choice of b_x is smaller than the optimal point. It turned out that for the case $c_p < -0.5$ the choice $b_x = 2\,x_i$ leads to better fitting hats. Note that in both cases it is not necessary to use the exact point x_i. A rough approximation with a relative error not above, e.g., 0.01 is enough. This is important as in most cases x_i must be determined by a numerical search.

3.3 Selecting parameters c_p and c_t and the design points

Theorem 5 shows that $\mathrm{ilc}_f(x_i) < -1$ cannot happen. But what about the case where $\mathrm{ilc}_f(x_i) = -1$? Applying the standard reasoning of TDR this implies that we have no chance to obtain a T_c-concave density on (b_x, ∞) or a T_c-concave inverse density on $(0, b_x)$ and thus we cannot use standard TDR with several design points. Nevertheless, it is possible to construct a valid TDR-hat with one point of contact. Let us assume that in a design point $x_0 > b_x$, we construct for a fixed $c \leq \mathrm{lc}(x_0)$ a TDR hat function that touches the density f in x_0. As we do not assume that f is T_c-concave we have to check the correctness of the hat function. The below theorem shows that it suffices to check that the hat is above the density in the two endpoints of the interval b_l and b_r.

THEOREM 6. *Let $f(x)$ be a twice continuously differentiable density on the (possibly half-infinite open) interval (b_l, b_r); for a fixed $c > -1$ assume that the transformed density $T_c(f(x))$ has no more than two inflection points and is concave between them. A hat $h(x)$ is constructed for f using TDR with a single point of contact x_0 with $b_l < x_0 < b_r$. Then we have $f(x) \leq h(x)$ for all x in (b_l, b_r) if and only if $f(b_l) \leq h(b_l)$ and $f(b_r) \leq h(b_r)$.*

PROOF. If h is a valid hat function then the condition trivially holds. Now assume the conditions $f(b_l) \leq h(b_l)$ and $f(b_r) \leq h(b_r)$ are satisfied. Then the design point x_0 cannot fall into a region where f is T_c-convex since otherwise one of these conditions fails. Thus h is a valid hat in the closed interval between the two inflection points i_l and i_r. As $T_c(f(x))$ is convex on (b_l, i_l) and $h(x) \geq f(x)$ for $x = i_l$ it also holds on the entire interval. The same is true for the interval (i_r, b_r). For the case where only one or no inflection point exists we just set $i_l = b_l$ or $i_r = b_r$ or both and can then apply the same argument. □

It is not difficult to see that for an arbitrary c the inflection points of the transformed density $T_c(f(x))$ can be characterized by the equation $\mathrm{lc}_f(x) = c$. Thus if the local concavity lc_f has no local minimum in the interior of the domain, then for any fixed but arbitrary c the transformed density can never have more than two inflection points and is guaranteed to be concave between them. Hence we have proven:

COROLLARY 1. *Let $f(x)$ be a twice continuously differentiable density on the (possibly half-infinite open) interval (b_l, b_r). If $\mathrm{lc}_f(x)$ has no local minimum in the interior of (b_l, b_r), then a TDR hat function $h_c(x)$ constructed in an arbitrary point $b_l \le x_0 \le b_r$ satisfies $h(x) \ge f(x)$ for all $x \in (b_l, b_r)$ if and only if $h(b_l) > f(b_l)$ and $h(b_r) > f(b_r)$.*

Corollary 1 motivates the formulation of the following general condition on densities.

CONDITION 1. *The local concavity lc_f and the inverse local concavity ilc_f have no local minimum on the respective regions (b_x, ∞) and $(0, b_x)$.*

This condition implies by Corollary 1 that in order to prove the correctness of the hat in the pole region it suffices to check that the hat is correct for x very close to 0 and for $x = b_x$. For the tail region we have to check the hat for $x = b_x$ and for a very large x.

3.3.1 *The pole region.* As the density has a pole at 0, the hat for the pole region can only be correct if $c_p \le \lim_{x\to 0} \mathrm{ilc}_f(x)$. To get an approximate value for that limit it is useful to observe that this limit equals $\lim_{x\to 0} \log(f(x))/\log(x)$ by l'Hôpital's rule. Hence $\lim_{x\to 0} \mathrm{ilc}_f(x) \approx \log(f(x_0))/\log(x_0)$ for a value of x_0 very close to 0. Selecting $x_0 \leftarrow 10^{-8} x_i$ proved to be numerically acceptable for that task in our experiments. We use $c_p = \lim_{x\to 0} \mathrm{ilc}_f(x)$ as the first candidate for c_p.

For the optimal selection of the design point x_p we can apply Thm. 2 as we have constructed h_p by means of a hat to the inverse density $f^{-1}(y)$ consisting of a constant center and a tail part, see Figure 1. By this construction the formula for the optimal design point reduces to

$$x_p = b_x \, (1 + c_p)^{-1/c_p} \; . \tag{9}$$

We have to check the validity of the constructed hat $h_p(x)$ by testing the condition $h_p(x) > f(x)$ for $x = b_x$ and for x very close to zero (e.g. $x = 10^{-100}$). If this condition is violated in either of these two points we have to replace c_p by a value closer to -1 and then recalculate x_p and check the hat again. This procedure can be repeated till the hat is valid.

3.3.2 *The Tail Region.* For the tail region ($x > b_x$) we can calculate the optimal design point x_t using Thm. 1. It is very practical here that the optimal design point does not depend on c_t as this implies that we can select c_t afterwards. For finding the optimal x_t we solve $(x_t - b_x)\, f'(x_t) + f(x_t) = 0$ numerically (note that the procedure for computing x_i above is exactly the same with b_x replaced by 0 there).

The value for c_t should be as large as possible to obtain the smallest possible area below the hat. For many distributions the infimum of $\mathrm{lc}_f(x)$ for the tail region was achieved at b_x. So for these distributions an easy "conservative" choice is $c_t = \mathrm{lc}_f(b_x)$. However, it turned out that this choice leads to unnecessary high tails and large areas below the hat especially for densities with "heavy" tails. We have also seen above that it is even possible that $\mathrm{lc}_f(b_x) = -1$. We therefore should select a larger value for c_t. But it is clear that the largest possible value for c_t must be smaller than $\mathrm{lc}_f(x_t)$. If we assume that $\mathrm{lc}_f(x)$ is monotonically increasing we can get a good initial guess for c_t using $c_t = \mathrm{lc}_f(b_x)/2 + \mathrm{lc}_f(x_t)/2$. For any valid

hat we know that we need $c_t \le \lim_{x\to\infty} \mathrm{lc}_f(x)$. So it is a good start to use

$$c_t = \min(\mathrm{lc}_f(b_x)/2 + \mathrm{lc}_f(x_t)/2, \lim_{x\to\infty} \mathrm{lc}_f(x)) \,.$$

Note that $\lim_{x\to\infty} \mathrm{lc}_f(x)$ equals $\lim_{x\to\infty} \log(x)/\log(f(x))$ by l'Hôpital's rule. Hence $\lim_{x\to\infty} \mathrm{lc}_f(x) \approx \log(x_\infty)/\log(f(x_\infty))$ for a large value of x_∞.

To check whether the choice of c_t leads to a valid hat we know that it suffices to check the validity of the hat for $x = b_x$ and for a very large x. We found (perhaps to the surprise of some readers) that $1000\,x_i$ is a "good" approximation for ∞ here as most densities and their hats decrease fast and thus are often both rounded to zero for very large values of x. If the validity test for the hat fails we have to retry with a smaller value for c_t (i.e., closer to -1) and then make the check again till the hat is valid.

3.4 Performance Bounds

The new algorithm does not require knowledge about the order of the pole or the tail. Instead the behavior is estimated by calculating the concavity of the density in just three points and by checking the correctness of the constructed hats at the borders of the domains. The simple Condition 1 is enough to guarantee that our algorithm constructs valid hat functions. However, this does not necessarily imply that ITDR is always able to construct a hat function. Consider for example a super heavy-tailed density with a tail proportional to $1/(x\,(\log x)^2)$. The local concavity of such a density converges to -1 when x tends to infinity. Thus no hat function that is constructed using a transformation T_c has bounded integral and ITDR is not able to construct a valid hat function. The same holds for a pole proportional to $1/(x\,(\log x)^2)$. If we try ITDR for such a density the iterative procedure of retrying c-values closer and closer to -1 will never stop. So summarizing the set-up of the new algorithm is able to construct a valid generator when the local concavities $\mathrm{lc}_f(x)$ and $\mathrm{ilc}_f(x)$ have no local minima and their limits to infinity and 0 respectively are smaller than minus one.

For ITDR as for any other rejection algorithm we cannot expect that the rejection constant (i.e., the expected number of trials to generate one variate) is uniformly bounded for all monotone densities. We necessarily must have $\lim_{x\to 0} \mathrm{ilc}_f(x) > -1$ and $\lim_{x\to 0} \mathrm{lc}_f(x) > -1$. In that case it is even possible to give a general bound for the performance of ITDR:

THEOREM 7. *Let f be a monotone density on $(0,\infty)$ that fulfills Condition 1. If*

$$\mathrm{ilc}_f(x) \ge c_1 > -1 \text{ on } (0, b_x) \text{ and } \mathrm{lc}_f(x) \ge c_2 > -1 \text{ on } (b_x, \infty)$$

then the rejection constant is bounded by

$$\alpha = \frac{A_h}{A_f} \le \max\left((1+c_1)^{1/c_1}, \frac{1}{1 - 1/(1+c_2)^{1+1/c_2}}\right) \quad \text{for } -1 < c_1, c_2 < 0\,.$$

If c_1 or c_2 is 0 we have to replace the above expressions by their respective limits e *and* $\frac{\mathrm{e}}{\mathrm{e}-1}$.

PROOF. As the area A_f below the density cannot be larger than one for the pole region and for the tail region the result is a direct consequence of Thms. 1 and 2. □

4. IMPLEMENTATION AND COMPUTATIONAL EXPERIENCE

4.1 The Algorithm

All details of an algorithm that utilizes this simple approach to select the parameters together with the random variate generation are presented as Algorithm ITDR. There are some remarks concerning the implementation in a real world computer:

—The symbols $\tilde{f}(x)$ and $\tilde{f}'(x)$ denote the transformed density $\tilde{f}(x) = T_c(f(x))$ and its derivative, respectively.

—The transformation T_c and its derived functions are given as $T_c(x) = -x^c$, $T_c'(x) = -c\,x^{c-1}$, $T_c^{-1}(x) = (-x)^{1/c}$, $F_{T_c}(x) = -\frac{c}{c+1}\,(-x)^{(c+1)/c}$, and $F_{T_c}^{-1}(x) = -(-x(c+1)/c)^{c/(c+1)}$.

—The case $c = -1/2$ with $T_{-1/2}(x) = -1/\sqrt{x}$ is computational much faster as we have $T_{-1/2}^{-1}(x) = 1/x^2$, $F_{T_{-1/2}} = -1/x$, and $F_{T_{-1/2}}^{-1} = -1/x$. Thus if $c < -1/2$ is replaced by $c = -1/2$ the resulting hat is larger than the optimal hat but the generation time can be much faster (depending on the expenses of computing f).

—The computation of the logarithms of densities and their derivatives is much easier for many distributions than the direct computation of the density. Moreover, the algorithm becomes more stable as numerical under/overflow and serious round-off errors near the pole or for large values of x are less likely. Notice that $\mathrm{lc}_f(x) = (1/(\log(f(x))'))'$ and $\mathrm{ilc}_f(x) = 1 + x\,[\log(f(x))' + \log(f(x))''/\log(f(x))']$.

4.2 Densities with Bounded Domain

It is not difficult to adapt Algorithm ITDR such that it becomes applicable to unbounded densities on bounded domains $(0, b_r)$. If the tail is short it is possible to use only the pole part by setting $b_x = b_r$. Otherwise, we have to adapt the setup and the sampling algorithm for the tail part such that it works for a bounded domain; as the tail part is a standard TDR algorithm we can just follow the general principles explained by Hörmann et al. [2004, Chap. 4] and Hörmann [1995].

4.3 Checking the Correctness of the Hat During Sampling

If the density f is numerically unfriendly it may be difficult in practice to check Condition 1. As a simple alternative it is possible to check the correctness of the constructed hat during drawing samples. To do so we check for each value X generated from the hat distribution whether $h(X) \geq f(X)$. This validity check is very simple and for moderate to large sample sizes this method will certainly detect significant deviations between the correct distribution and the generated random variates. To be fully sure or if only small samples are necessary there is no alternative to checking the condition.

4.4 Computational Experiences

We tested our algorithm for the Gamma, Beta, F, Planck and Beta prime distributions for shape parameters $0.01 \leq a \leq 0.99$ and several different values for shape parameter b for the two Beta distributions. (Note that for these five distribution families the local concavity and the inverse local concavity have no local minimum and thus Condition 1 is fulfilled.) In all our experiments the rejection constant was

Algorithm 1 ITDR

Require: Monotone density $f(x)$ on $(0, \infty)$ with pole at 0 and its derivative $f'(x)$.
Ensure: Random variate X with density f.

/* Setup: candidate for b_x */
Find point x_i approximately satisfying $x_i\, f'(x_i) + f(x_i) = 0$.
/* Setup: pole region */
Set $c_p \leftarrow \min(0, \log(f(x_i\, 10^{-8}))/\log(x_i\, 10^{-8}))$.
if $c_p < -0.5$ set $b_x \leftarrow 2\, x_i$ **else** $b_x \leftarrow x_i$.
Set $x_p \leftarrow b_x\, (1 + c_p)^{-1/c_p}$.
Set $\beta_p \leftarrow T'_{c_p}(x_p)/f'(x_p)$ and $\alpha_p \leftarrow T_{c_p}(x_p) - \beta_p\, f(x_p)$.
if $h_p(10^{-100}) < f(10^{-100})$ or $h_p(b_x) < f(b_x)$ **then** /* $h_p(x) = (T_{c_p}(x) - \alpha_p)/\beta_p$ */
  Set $c_p \leftarrow 0.9\, c_p - 0.1$ and goto Step 4.
/* Setup: tail region */
Find point x_t approximately satisfying $(x_t - b_x)\, f'(x_t) + f(x_t) = 0$.
Set $c_t \leftarrow \min(\mathrm{lc}_f(b_x)/2 + \mathrm{lc}_f(x_t)/2, \log(x_i\, 10^6)/\log(f(x_i\, 10^6)))$.
Compute and store $\tilde{f}(x_t) \leftarrow T_{c_t}(f(x_t))$ and $\tilde{f}'(x_t) \leftarrow T'_{c_t}(f(x_t))\, f'(x_t)$.
/* $h_t(x) = T^{-1}_{c_t}(\tilde{f}(x_t) + \tilde{f}'(x_t)(x - x_t))$ */
if $h_t(b_x) < f(b_x)$ or $h_t(1000\, b_x) < f(1000\, b_x)$ **then**
  Set $c_t \leftarrow (c_t + \mathrm{lc}_f(b_x))/2$ and goto Step 10.
/* Setup: parameters */
Set $b_y \leftarrow h_p(b_x)$.
Set $A_p \leftarrow -F_{T_{c_p}}(\alpha_p + \beta_p\, b_y)/\beta_p$, $\quad A_c \leftarrow b_y\, b_x$, and
$A_t \leftarrow -F_{T_{c_t}}(\tilde{f}(x_t) + \tilde{f}'(x_t)(b_x - x_t))/\tilde{f}'(x_t)$.
/* For the definition of $F_T(x)$ and $F_T^{-1}(x)$ see Eq. (4) */
Set $A_{tot} \leftarrow A_p + A_c + A_t$. /* area below hat */
/* Generator */
loop
  Generate $U \sim \mathcal{U}(0, A_{tot})$. /* uniform distribution on $(0, A_{tot})$ */
  if $U < A_p$ **then** /* pole region */
    Set $Y \leftarrow (F^{-1}_{T_{c_p}}(\beta_p\, U + F_{T_{c_p}}(\alpha_p + \beta_p\, b_y)) - \alpha_p)/\beta_p$.
    Generate $X \sim \mathcal{U}(0,\, T^{-1}_{c_p}(\alpha_p + \beta_p\, Y))$.
  else if $U < A_p + A_c$ **then** /* central region */
    Set $U \leftarrow U - A_p$.
    Set $X \leftarrow U\, b_x/A_c$.
    Generate $Y \sim \mathcal{U}(0, b_y)$.
  else /* tail region */
    Set $U \leftarrow U - (A_p + A_c)$.
    Set $X \leftarrow x_t + (F^{-1}_{T_{c_t}}(\tilde{f}'(x_t)\, U + F_{T_{c_t}}(\tilde{f}(x_t) + \tilde{f}'(x_t)\, (b_x - x_t))) - \tilde{f}(x_t))/\tilde{f}'(x_t)$.
    Generate $Y \sim \mathcal{U}(0,\, T^{-1}_{c_t}(\tilde{f}(x_t) + \tilde{f}'(x_t)\, (x - x_t)))$.
  if $Y \le f(X)$ **then** /* accept */
    return X.

below 1.1 which indicates that the hat fits very well for all distributions. The speed of the algorithm is approximately the same for all distributions we tried, about ten times slower than the generation of one exponential random variate by inversion.

But also the many special algorithms for the Gamma, the Beta and the Beta prime distribution (see [Devroye 1986] for an overview) are much slower than the very simple generation of an exponential random variate. In a small time comparison for the gamma distribution among three different special algorithms of [Devroye 1986] one was 20 percent slower, one was about the same speed and one was 20 percent faster than inverse TDR in the fixed parameter situation. In the changing parameter case inverse TDR is about four times slower but this is not astonishing as an automatic algorithm of course requires a slower set-up than a special algorithm. For the Planck distribution we found only one algorithm in the literature [Devroye 1986]. It requires the generation of Zipf and Gamma variates and is clearly slower than our universal algorithm in the fixed parameter situation.

4.5 Numerical Stability

It is clear that a rejection algorithm for an unbounded density may lead to numerical problems. Of course any representation of real numbers on a computer can only contain a discrete subset. But the usual floating point arithmetic used today contains a discrete subset that is by far most dense around 0 [Overton 2001]. The smallest floating point number larger than 0 is approximately 10^{-320} whereas the smallest floating point number larger than 1 is just about $1 + 10^{-16}$. This is the reason why in this paper we have always considered the pole to be located at 0. If this is not the case and the pole of the random variate X is located at x_0 instead it is safer to code the density $f_Y(y)$ of $Y = X - x_0$ and to generate variates Y first that are then transformed back using $X = Y + x_0$. We have to be careful here when coding the density. Just plugging in the definition $f_Y(y) = f(y + x_0)$ may lead to a problem as for very small y we may loose a lot of precision when adding the comparatively large number x_0.

We have experienced similar problems when applying our algorithm to the Planck distribution which has a density proportional to $x^a/(e^x - 1)$. When we used just the naive implementation of this density and its derivative the setup was not able to construct a hat function because of the rounding errors due to extinction for x close to 0. To fix the problem it suffices to replace $(e^x - 1)$ by the first three terms of its Taylor series expansion at 0 for the case where $x < 10^{-5}$.

We ran χ^2-goodness-of-fit tests with sample sizes 10^6 and 10^7 on the output of our algorithm for all five distributions from Sect. 4.4. The results were satisfactory for all four distributions when the first parameter a was above 0.02. For smaller values of a the test started to show problems.

The reason for that problem can be explained easily: First we have to observe that, like any other rejection algorithm, our algorithm cannot generate values exactly equal to 0. (As $f(0)$ and $h(0)$ are infinity a rejection algorithm cannot decide to accept or reject 0 and so it is automatically excluded by most floating point units as $(\infty < \infty)$ is naturally not considered as true.) As we have stated above the smallest positive floating point number representable on our computer is close to 10^{-320}. Assume the density $f(x, a) = a\,x^{a-1}$ on $(0, 1)$ and its corresponding CDF $F(x, a) = x^a$ which is — for small values of x and a — very similar to the densities

of all five distributions we tested. We then find $F(10^{-320}, 0.01) = 10^{-3}/2$. Thus a non-negligible part of the pole is cut off by the rejection step of the algorithm after these numbers had been rounded to 0 by the procedures of the floating point arithmetic. This problem makes the χ^2-test fail. As mentioned the same problem exists for any rejection algorithm. A possible way out of the problem for a distribution with known CDF is to define a mass point at 0 with its probability equal to $P(X < 10^{-320})$ (see [Ahrens 1995]) but of course this probability is unknown if we just know the density and not the CDF.

5. CONCLUSIONS

We have introduced the *inverse transformed density rejection* method and designed an algorithm that works for most monotone unbounded densities. The density and its derivative must be available together with the location of the pole. The order of the pole is estimated numerically in the setup of the algorithm. Simple conditions on the density were derived that guarantee that the setup constructs a valid hat function. When it is difficult or too time consuming to verify these conditions, it is also possible to check the validity of the constructed algorithm during the sampling procedure. Numerical experiments indicate that the algorithm is working correctly and moderately fast for many different unbounded densities.

ACKNOWLEDGMENTS

The authors want to thank the area editor and the referees for their help in improving the presentation of the paper.

REFERENCES

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

Box, G. E. P. and Cox, D. R. 1964. An analysis of transformations. *Journal of the Royal Statistical Society, B 26*, 211–243.

Cheng, R. C. H. and Feast, G. M. 1980. Gamma variate generators with increased shape parameter range. *Commun. ACM 23*, 389–394.

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

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

Hörmann, W. 1995. A rejection technique for sampling from T-concave distributions. *ACM Trans. Math. Software 21,* 2, 182–193.

Hörmann, W. and Derflinger, G. 1996. Rejection-inversion to generate variates from monotone discrete distributions. *ACM Trans. Model. Comput. Simul. 6,* 3, 169–184.

Hörmann, W., Leydold, J., and Derflinger, G. 2004. *Automatic Nonuniform Random Variate Generation*. Springer-Verlag, Berlin Heidelberg.

Leydold, J. and Hörmann, W. 2000–2006. *UNU.RAN – A Library for Universal Non-Uniform Random Variate Generation*. Department of Statistics and Mathematics, WU Wien, Vienna, Austria. available at `http://statmath.wu-wien.ac.at/unuran/`.

Overton, M. L. 2001. *Numerical Computing with IEEE Floating Point Arithmetic*. SIAM, Philadelphia.