

A Generalized Transformed Density Rejection Algorithm

Hörmann, Wolfgang; Leydold, Josef

DOI:

[10.57938/d3bf5f14-9b27-4f79-8014-59667addc9fa](https://doi.org/10.57938/d3bf5f14-9b27-4f79-8014-59667addc9fa)

Published: 01/05/2022

Document Version

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Hörmann, W., & Leydold, J. (2022). *A Generalized Transformed Density Rejection Algorithm*. WU Vienna University of Economics and Business. Research Report Series / Department of Statistics and Mathematics No. 135 <https://doi.org/10.57938/d3bf5f14-9b27-4f79-8014-59667addc9fa>

A Generalized Transformed Density Rejection Algorithm

Wolfgang Hörmann, Josef Leydold

Research Report Series
Report 135, May 2022

Institute for Statistics and Mathematics
<http://statmath.wu.ac.at/>



A Generalized Transformed Density Rejection Algorithm

Wolfgang Hörmann and Josef Leydold

Abstract Transformed density rejection is a very flexible method for generating non-uniform random variates. It is based on the acceptance-rejection principle and utilizes a strictly monotone map that transforms the given density into a concave or convex function. Hat function and squeezes are then constructed by means of tangents and secant. We present a new method that works for arbitrary one time continuously differentiable densities. It requires together with the log-density and its derivative a partition of the domain into subdomains that contain at most one inflection point. This improves a previous method of the authors in which also the second derivative is required. We show how the algorithm can be applied to generate from the Generalized Inverse Gaussian distribution, from the Generalized Hyperbolic distribution and from the Watson distribution. The new algorithm can also generate random variates from truncated distributions without problems.

Key words: non-uniform random variate generation, black-box algorithm, transformed density rejection, adaptive rejection sampling

W. Hörmann
Department of Industrial Engineering
Boğaziçi University
34342 Bebek-İstanbul, Turkey
e-mail: hormannw@boun.edu.tr

J. Leydold
Institute for Statistics and Mathematics
Vienna University of Economics and Business
Welthandelsplatz 1, Building D4, 1020 Vienna, Austria
e-mail: josef.leydold@wu.ac.at

1 Introduction

Monte Carlo methods and stochastic simulation are very powerful tools for computing metric values in models. A crucial step is the sampling of uniform random numbers and non-uniform random variates. For the latter acceptance-rejection sampling is often used. Then an upper bound h (called *hat* function) and optionally a lower bound s (called *squeeze*) has to be found that satisfy $0 \leq s(x) \leq f(x) \leq h(x)$. Hat function h must be some multiple of a density function that allows for easy sampling from (typically by inversion) and squeeze s may be used to reduce the computational expense of evaluating f . Once values of h and s have been found, to generate a value of X from a distribution with density f , the following steps are necessary:

1. Generate a random variate X with density proportional to h .
2. Generate a $(0, 1)$ uniform random number, U .
3. If $U h(X) \leq s(X)$, then return X .
4. If $U h(X) \leq f(X)$, then return X .
5. Otherwise, try again.

Although executing the five steps above is simple, the challenge in implementing acceptance-rejection sampling is in finding appropriate functions h and s . There exist many papers proposing such functions especially tailored for standard distributions, see, e.g., [9] for an extensive survey.

Devroye [8] discussed a different approach and proposed a general method to construct hat functions that works for all distributions with log-concave densities. Notice here that the given density need not be normalized. That is, any multiple of a density (with unknown proportionality factor) can be used. Gilks and Wild [11] use tangents and secants of the log-density to construct hat and squeeze functions, resp. Thus the hat distribution is a mixture of truncated exponentially distributed random variates with disjoint domains. Hence sampling by inversion is fast and simple. The interval boundaries are computed as the intersection points of the tangents. The initial subdivision is then refined by adaptive rejection sampling (ARS).

Hörmann [12] generalized this idea for the class of T -concave distributions. A density f is called T -concave if the transformed density $\tilde{f} = T \circ f$ is concave, where $T: (0, \infty) \rightarrow \mathbb{R}$ is a differentiable and monotonically increasing transformation. If f is T -concave, the tangent $\tilde{t}(x) = \alpha + \beta x$ to \tilde{f} is greater than \tilde{f} for all x in the domain of f , making the function $t(x) = T^{-1}[\tilde{t}(x)] = T^{-1}(\alpha + \beta x)$ a hat function to f . He also suggested a class of Box-Cox-like transformations where again sampling from the hat distribution by inversion is quite cheap, see Table 1. We want to note here that a T_c -concave density is also T_{c_1} -concave for every $c_1 \leq c$. Furthermore, we need $c > -1$ for unbounded intervals as otherwise the integral of the hat function is unbounded. For a detailed discussion we refer to [13].

Similarly, if f is T -concave, the secant to the transformed density, \tilde{r} , can be used to construct the squeeze function, s , for the density in a given interval. Evans and Swartz [10] show that the opposite applies (in that tangents are used to construct

Table 1 The family T_c of transformations. F_T denotes the antiderivative of T_c^{-1}

c	$T_c(x)$	$T_c^{-1}(x)$	$F_T(x)$	$F_T^{-1}(x)$
> 0	x^c	$x^{1/c}$	$\frac{c}{c+1}x^{(c+1)/c}$	$\left(\frac{c+1}{c}x\right)^{c/(c+1)}$
0	$\log(x)$	e^x	e^x	$\log(x)$
< 0	$-x^c$	$(-x)^{1/c}$	$-\frac{c}{c+1}(-x)^{(c+1)/c}$	$-\left(-\frac{c+1}{c}x\right)^{c/(c+1)}$
$-1/2$	$-1/\sqrt{x}$	$1/x^2$	$-1/x$	$-1/x$
-1	$-1/x$	$-1/x$	$-\log(-x)$	$-\exp(-x)$

the squeeze function and secants are used to construct the hat function) when f is T -convex.

It is quite obvious that this approach works for an arbitrary distribution with differentiable density when we partition its domain into intervals where the density is either T -concave or T -convex. For unbounded sub-intervals it is required that the density is T -concave. Although it is not necessary to use the same transformation in each subdomain, identifying these intervals requires the inflection points of \tilde{f} , which are often difficult to obtain.

For that reason Botts [5] relaxed the requirement of knowing the exact position of the inflection points. He proposes a method requiring a subdivision into intervals where the transformed density is either concave, convex, or has exactly one inflection point. For the latter case, he introduces an additional transformation and compiles a new sampling algorithm.

This idea has been simplified by [6]. It avoids the necessity of an additional transformation and relaxes the rather strong properties of the densities. Instead the user has to provide an implementation of the second derivative of the (log-) density. A ready-to-use version of the proposed algorithm is provided as **R** package `Tinflex`, see [14].

A closer look at the `Tinflex` algorithm reveals that it is enough to know the *sign* of the second derivative. In this contribution we thus show how to develop a method that only requires the transformed density and its derivative for constructing hat and squeeze in a reliable way at the expense of a slightly increased complexity of the setup part of the generator. We are convinced that from a mathematical point of view it is clear that an algorithm with less input is more elegant and preferable. Also we can assume that most users will be glad to have to supply only two functions instead of three. It is certainly true that numeric or automatic derivation could be used to reach that aim of more convenience for the user. But of course that depends also on the experience the user has with software for numeric or automatic derivation and is able to use it correctly. And there remains the fact that for numerically difficult densities like, e.g., for the generalized hyperbolic (GH) distribution (see Section 5.1) the computational burden of the set-up step is reduced by removing the necessity of the second derivative. It is also likely that for such numerically difficult densities the new version of `Tinflex` is numerically more robust. In addition this approach allows for a more user-friendly implementation of the sampling algorithm. It works

also without problem for generating very fast from densities truncated to an arbitrary interval. This is not a simple task as exhibited in [3].

The paper is organized as follows: In Section 2 we shortly summarize the sampling method from [6]. In Section 3 we present the proposed improvement. Section 4 compiles the entire algorithm, and in Section 5 we demonstrate how to apply the new algorithm to generate from the Generalized Inverse Gaussian distribution, from the Generalized Hyperbolic distribution and from the Watson distribution.

2 Transformed Density Rejection with Inflection Points

In this section we summarize the method of [6] and restate the main result in Theorem 1 that is more suitable for our purpose. Moreover, a review of the given proofs show that the conditions for density f can be relaxed as following.

- (C1) Density f and thus transformed density \tilde{f} are continuous and piece-wise twice continuously differentiable.
- (C2) There is only a finite number of points where \tilde{f}'' does not exist. Around each of these points either \tilde{f}' is monotone or \tilde{f}'' changes sign. This excludes transformed densities like $\tilde{f}(x) = e^{-|x|}$ while $\tilde{f}(x) = \sqrt[3]{x}$ does work.
- (C3) We are given a partition of the domain with finitely many breaking points $-\infty \leq b_0 < b_1 < \dots < b_n < b_{n+1} \leq \infty$ where the following holds:
 - (C3a) In each bounded interval $[b_i, b_{i+1}]$ of the partition the closures of the sets $\{x: \tilde{f}''(x) \leq 0\}$ and $\{x: \tilde{f}''(x) \geq 0\}$ are connected or empty.
 - (C3b) In each unbounded interval $(-\infty, b_1]$ or $[b_{n-1}, \infty)$, \tilde{f} must be concave and strictly monotone.

Observe that Condition (C3a) holds when there is at most one inflection point in $[b_i, b_{i+1}]$ as stated in the original paper. However, (C3a) also allows transformed densities which are linear on subdomains. Another consequence is that there exists a point $y^* \in (b_l, b_r)$ that separates subdomains $[b_l, y^*]$ and $[y^*, b_r]$ where \tilde{f} is convex and concave, resp., whenever both sets are non-empty. Condition (C3b) is required as otherwise we cannot create a hat function with bounded integral.

Now let $[b_l, b_r]$ be an interval in the domain of density f . We denote the tangent of the transformed density \tilde{f} in the boundary points by $\tilde{t}_l(x) = \tilde{f}(b_l) + \tilde{f}'(b_l)(x - b_l)$ and $\tilde{t}_r(x) = \tilde{f}(b_r) + \tilde{f}'(b_r)(x - b_r)$, resp. Its secant is denoted by $\tilde{r}(x)$ with slope

$$R = \frac{\tilde{f}(b_r) - \tilde{f}(b_l)}{b_r - b_l}. \quad (1)$$

In general we use a tilde $\tilde{}$ to denote functions in transformed scale.

The algorithm is based on the following proposition that immediately follows from [6, Thms. 1 and 2].

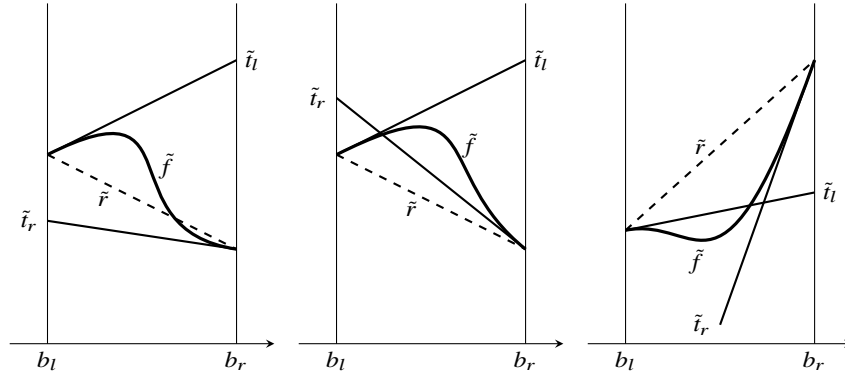


Fig. 1 Intervals of types (Ia), (IIa), and (IIIa) from Table 2

Theorem 1 Let $[b_l, b_r]$ be a bounded closed interval where \tilde{f} satisfies Condition (C3a). Then it belongs to one of the eight types that are listed in Table 2. There are no other types.

The inequalities in Table 2 allow to determine the type of the interval and to create hat function and squeeze. For types (IVa) and (IVb) both tangents can be used. In [6] the one where \tilde{f} is larger in the respective construction points b_l and b_r is proposed. Figure 1 illustrates three of the possible types.

Table 2 Types of intervals $[b_l, b_r]$

Type	\tilde{f}' and R	\tilde{f}''	squeeze and hat
Ia	$\tilde{f}'(b_l), \tilde{f}'(b_r) \geq R$	$\tilde{f}''(b_l) \leq 0 \leq \tilde{f}''(b_r)$	$\tilde{t}_r(x) \leq \tilde{f}(x) \leq \tilde{t}_l(x)$
Ib	$\tilde{f}'(b_l), \tilde{f}'(b_r) \leq R$	$\tilde{f}''(b_l) \geq 0 \geq \tilde{f}''(b_r)$	$\tilde{t}_l(x) \leq \tilde{f}(x) \leq \tilde{t}_r(x)$
IIa	$\tilde{f}'(b_l) \geq R \geq \tilde{f}'(b_r)$	$\tilde{f}''(b_l) \leq 0 \leq \tilde{f}''(b_r)$	$\tilde{r}(x) \leq \tilde{f}(x) \leq \tilde{t}_l(x)$
IIb	$\tilde{f}'(b_l) \geq R \geq \tilde{f}'(b_r)$	$\tilde{f}''(b_l) \geq 0 \geq \tilde{f}''(b_r)$	$\tilde{r}(x) \leq \tilde{f}(x) \leq \tilde{t}_r(x)$
IVa	$\tilde{f}'(b_l) \geq R \geq \tilde{f}'(b_r)$	$\tilde{f}''(b_l), \tilde{f}''(b_r) \leq 0$	$\tilde{r}(x) \leq \tilde{f}(x) \leq \tilde{t}_l(x), \tilde{t}_r(x)$
IIIa	$\tilde{f}'(b_l) \leq R \leq \tilde{f}'(b_r)$	$\tilde{f}''(b_l) \leq 0 \leq \tilde{f}''(b_r)$	$\tilde{t}_r(x) \leq \tilde{f}(x) \leq \tilde{r}(x)$
IIIb	$\tilde{f}'(b_l) \leq R \leq \tilde{f}'(b_r)$	$\tilde{f}''(b_l) \geq 0 \geq \tilde{f}''(b_r)$	$\tilde{t}_l(x) \leq \tilde{f}(x) \leq \tilde{r}(x)$
IVb	$\tilde{f}'(b_l) \leq R \leq \tilde{f}'(b_r)$	$\tilde{f}''(b_l), \tilde{f}''(b_r) \geq 0$	$\tilde{t}_l(x), \tilde{t}_r(x) \leq \tilde{f}(x) \leq \tilde{r}(x)$

Remark 1 Notice:

- Transformed density \tilde{f} is concave near b_l in types (Ia), (IIa), (IIIa) and (IVa) and convex otherwise. It is concave near b_r in types (Ib), (IIb), (IIIb), and (IVa) and convex otherwise.
- The types in Table 2 are determined by R and the values of \tilde{f}' and \tilde{f}'' at boundary points b_l and b_r .

- Tangents \tilde{t}_l and \tilde{t}_r do not intersect in $[b_l, b_r]$ if and only if the interval is of type (Ia) or (Ib).
- Tangents \tilde{t}_l and \tilde{t}_r and secant \tilde{r} form a triangle in $[b_l, b_r]$ with the intersection point of the tangents above the secant if and only if the interval is of type (IIa), (IIIb), or (IVa); and below if and only if the interval is of type (IIIb), (IIIa), or (IVb).
- In intervals of type (IVa) and (IVb), \tilde{f}'' cannot change sign and thus \tilde{f} is concave and convex, resp., in $[b_l, b_r]$.

It is then straightforward to construct a sampling routine on interval $[b_l, b_r]$. Let

$$\tilde{h} = \alpha + \beta(x - x_0),$$

be the linear hat function for \tilde{f} where x_0 is either b_l or b_r . When the tangent serves as the hat function to \tilde{f} , $\alpha = \tilde{f}(x_0)$ and $\beta = \tilde{f}'(x_0)$, and when the secant serves as the hat function to \tilde{f} , $\alpha = \tilde{f}(x_0)$ and $\beta = R$. The area below the hat then becomes

$$\begin{aligned} A_h &= \int_{b_l}^{b_r} h(x) dx = \int_{b_l}^{b_r} T_c^{-1}(\alpha + \beta(x - x_0)) dx \\ &= \frac{1}{\beta} [F_T(\alpha + \beta(b_r - x_0)) - F_T(\alpha + \beta(b_l - x_0))], \end{aligned} \quad (2)$$

where F_T denotes the anti-derivative of T_c^{-1} . The (non-normalized) CDF, $H(x)$, of the density proportional to h then is given by

$$\begin{aligned} H(x) &= \int_a^x T_c^{-1}(\alpha + \beta(t - x_0)) dt \\ &= \frac{1}{\beta} [F_T(\alpha + \beta(x - x_0)) - F_T(\alpha + \beta(a - x_0))] . \end{aligned}$$

Notice that $H(b_r) = A_h$. Thus the inverse $H^{-1}(u)$ for $u \in [0, A_h]$ is then given by

$$H^{-1}(u) = x_0 + \frac{1}{\beta} [F_T^{-1}(\beta u + F_T(\alpha + \beta(a - x_0))) - \alpha] . \quad (3)$$

Remark 2 When x_0 is very close to a maximum or another point where $\tilde{f}'(x_0) \approx 0$, then formula (3) becomes sensitive to numerical errors. Then one should replace the exact formula by approximations which are accurate up to the resolution of the floating point numbers, see [6] for details.

Remark 3 When $c \neq 0$ there might be a problem when using tangents to construct hat or squeeze functions. If \tilde{t} has a root in the corresponding interval, then \tilde{t} cannot be transformed back into a valid hat or squeeze function. It is then necessary to further subdivide the corresponding interval as described below.

The performance of an acceptance-rejection method can be measured by means of the rejection constant defined as the ratio of the area below the hat function

and the area below the density. It gives the expected number of iterations of the acceptance-rejection loop for getting one (accepted) random variate. An advantage of transformed density rejection as described here is that the rejection constant is bounded by the ratio

$$\rho = \frac{\text{area below hat}}{\text{area below squeeze}} . \quad (4)$$

It even allows to estimate the required number of (usually expensive) evaluations of \tilde{f} which is approximately given by $\rho - 1$, see [13]. If ρ is close to one than the marginal generation time hardly depends on the given density.

Another advantage of this method is that ρ can be made as close to 1 as requested by the user. Indeed for the twice continuously differentiable densities with bounded domain we find

$$\rho = 1 + O(1/N^2) \quad (5)$$

when we have N intervals of equal length, see [6].

There exist some methods to find non-overlapping intervals of the domain which result in arbitrary small values of $\rho - 1$. One starts with any subdivision that satisfies Condition (C3). Then this subdivision can be refined by means of adaptive rejection sampling (ARS) as proposed by [11] where rejected points are used to split the corresponding interval into two parts until the requested value of ρ is reached. Alternatively we can iteratively subdivide intervals where the area between hat and squeeze is above some threshold value during the setup, see [15]. We propose to use the ‘‘arc-mean’’ of the boundaries of interval (b_{i-1}, b_i) for splitting intervals:

$$p_{\text{arc}} = \tan\left(\frac{1}{2}(\arctan(b_{i-1}) + \arctan(b_i))\right) \quad (6)$$

where $\arctan(\pm\infty)$ is set to $\pm\pi/2$, see also [13, Sect. 4.4.6].

It is then straightforward to compile an algorithm based on the above principles. In Algorithm 1 in Sect. 3 below we present the entire algorithm that makes use of the proposed improvements.

3 Determine Signs of Second Derivatives

We can see from Table 2 that we need the *sign* of \tilde{f}'' at the two boundary points only in order to distinguish between types (IIa), (IIb) and (IVa) as well as between types (IIIa), (IIIb) and (IVb). There is no necessity to compute its exact value. So we propose a method that characterizes these types reliably when only the first derivative is available. Again the conditions of Sect. 2 must be satisfied. We discuss two cases:

Case 1: We subdivide an interval of known type and determine the types of the two subintervals. This is useful when the signs of \tilde{f}'' are given for the boundaries of the initial intervals and we have to split an interval in order to improve the hat and squeeze. This scenario seems plausible as the user already needs a rough estimate for the inflection points.

Case 2: No such information is given and we have to determine the type of the interval without evaluating the second derivative.

For this purpose we first summarize some basic properties of concave and convex functions. In particular we will make use of properties (a) and (b).

Lemma 1 *Let \tilde{f} be a piece-wise C^2 -function on domain $[b_1, b_2]$ such that Condition (C2) holds.*

- (a) *Point $p^* \in (b_1, b_2)$ is an inflection point of \tilde{f} if and only if p^* is an extremal point of its derivative \tilde{f}' .*
- (b) *Derivative \tilde{f}' is monotonically decreasing (increasing) if and only if \tilde{f} is concave (convex).*
- (c) *Function \tilde{f} is concave (convex) if and only if $\tilde{f}''(x) \leq 0$ ($\tilde{f}''(x) \geq 0$) for all $x \in [b_1, b_2]$.*
- (d) *Let $\tilde{t}(x)$ be a tangent of \tilde{f} .
If \tilde{f} is concave, then $\tilde{f}(x) \leq \tilde{t}(x)$ for all $x \in [b_1, b_2]$.
If \tilde{f} is convex, then $\tilde{f}(x) \geq \tilde{t}(x)$ for all $x \in [b_1, b_2]$.*
- (e) *Let $\tilde{t}(x) = \tilde{f}(x_0) + \tilde{f}'(x_0)(x - x_0)$ be the tangent in x_0 .
If $\tilde{f}(y) \leq \tilde{t}(y)$ for some $y > x_0$, then $\tilde{f}'(x_0) \geq (\tilde{f}(y) - \tilde{f}(x_0)) / (y - x_0)$.
If $\tilde{f}(y) \geq \tilde{t}(y)$ for some $y < x_0$, then $\tilde{f}'(x_0) \leq (\tilde{f}(y) - \tilde{f}(x_0)) / (y - x_0)$.*

By our assumptions the sign of the second derivative can be determined by triples of points as stated in the next proposition.

Lemma 2 *Let \tilde{f} be a piecewise C^2 -function on domain $[b_1, b_2]$ such that Conditions (C2) and (C3a) hold. Let $b_1 \leq p_1 < p_2 < p_3 \leq b_2$.*

- (a) *If $\tilde{f}'(p_2) \leq \min\{\tilde{f}'(p_1), \tilde{f}'(p_3)\}$, then $\tilde{f}''(p_1) \leq 0 \leq \tilde{f}''(p_3)$.
If $\tilde{f}'(p_2) \geq \max\{\tilde{f}'(p_1), \tilde{f}'(p_3)\}$, then $\tilde{f}''(p_1) \geq 0 \geq \tilde{f}''(p_3)$.*
- (b) *If $\tilde{f}'(p_1) \leq \tilde{f}'(p_2) \leq \tilde{f}'(p_3)$, then $\tilde{f}''(p_2) \geq 0$.
If $\tilde{f}'(p_1) \geq \tilde{f}'(p_2) \geq \tilde{f}'(p_3)$, then $\tilde{f}''(p_2) \leq 0$.*

Proof By our assumptions there is at most one point or interval where \tilde{f}' is extremal in the open interval (b_1, b_2) . If $\tilde{f}'(p_2) \leq \min\{\tilde{f}'(p_1), \tilde{f}'(p_3)\}$, then there is exactly one minimum (interval) of \tilde{f}' in (p_1, p_3) and hence \tilde{f}' is decreasing near p_1 and increasing near p_3 . Thus (a) follows.

If $\tilde{f}'(p_1) \leq \tilde{f}'(p_2) \leq \tilde{f}'(p_3)$, then \tilde{f}' is either monotonically increasing, or has a minimum in subinterval (p_1, p_2) , or has a maximum in (p_2, p_3) . In all cases \tilde{f}' is increasing around p_2 and hence $\tilde{f}''(x_2) \geq 0$ as claimed in (b). \square

These elementary tools now enable us to determine \tilde{f}'' at the boundary points of the intervals. In Sect. 3.1 we look at the case where no additional information is available. In Sect. 3.2 we split intervals of given types and determine the types of the corresponding subintervals.

3.1 Initial Intervals

Theorem 2 Let \tilde{f} be a piecewise C^2 -function on domain $[b_l, b_r]$ such that Conditions (C2) and (C3a) hold. Let $b_l < p < b_r$ be some point. Then one of the cases in Table 3 holds. No other cases are possible. The properties of the combined types in cases (3.3.3) and (4.3.3) are listed in Table 4.

Table 3 Determination of interval types by means of transformed density \tilde{f} and its derivative \tilde{f}' . Point $p \in (b_l, b_r)$ must be an interior point. Symbol (IIIa | IVb) + (IIIb | IVb) means that we have to split the interval at point p into the two subintervals $[b_l, p]$ and $[p, b_r]$

Case	\tilde{f}' and R	$\tilde{f}(p)$	Type
(1)	$\tilde{f}'(b_l), \tilde{f}'(b_r) \geq R$		(Ia)
(2)	$\tilde{f}'(b_l), \tilde{f}'(b_r) \leq R$		(Ib)
(3)	$\tilde{f}'(b_l) \geq R \geq \tilde{f}'(b_r)$		—
(3.1)	$\tilde{f}'(p) \leq \tilde{f}'(b_r)$		(IIa)
(3.2)	$\tilde{f}'(p) \geq \tilde{f}'(b_l)$		(IIb)
(3.3)	$\tilde{f}'(b_l) \geq \tilde{f}'(p) \geq \tilde{f}'(b_r)$		—
(3.3.1)		$\tilde{f}(p) > \tilde{t}_l(p)$	(IIb)
(3.3.2)		$\tilde{f}(p) > \tilde{t}_r(p)$	(IIa)
(3.3.3)		$\tilde{f}(p) \leq \tilde{t}_l(p), \tilde{t}_r(p)$	(IIb IVa) + (IIa IVa)
(4)	$\tilde{f}'(b_l) \leq R \leq \tilde{f}'(b_r)$		—
(4.1)	$\tilde{f}'(p) \leq \tilde{f}'(b_l)$		(IIIa)
(4.2)	$\tilde{f}'(p) \geq \tilde{f}'(b_r)$		(IIIb)
(4.3)	$\tilde{f}'(b_l) \leq \tilde{f}'(p) \leq \tilde{f}'(b_r)$		—
(4.3.1)		$\tilde{f}(p) < \tilde{t}_l(p)$	(IIIa)
(4.3.2)		$\tilde{f}(p) < \tilde{t}_r(p)$	(IIIb)
(4.3.3)		$\tilde{f}(p) \geq \tilde{t}_l(p), \tilde{t}_r(p)$	(IIIa IVb) + (IIIb IVb)

Proof Let $p^* \in [b_l, b_r]$ denote the possible inflection point of \tilde{f} (or one of the points that separate the subdomains where \tilde{f} is concave and convex, resp.).

Obviously one of cases (1), (2), (3), or (4) must hold. According to Table 2 cases (1) and (2) determine types (Ia) and (Ib), resp.

Case (3): We have $\tilde{f}'(b_l) \geq R \geq \tilde{f}'(b_r)$. Then by Table 2 interval $[b_l, b_r]$ is of type (IIa), (IIb), or (IVa). In order to distinguish between these cases we look at $\tilde{f}'(p)$. If $\tilde{f}'(p) \leq \tilde{f}'(b_r) = \min\{\tilde{f}'(b_l), \tilde{f}'(b_r)\}$ (case 3.1), then $\tilde{f}''(b_l) \leq 0 \leq \tilde{f}''(b_r)$ by Lemma 2(a) and thus $[b_l, b_r]$ is of type (IIa). Analogously, if $\tilde{f}'(p) \geq \tilde{f}'(b_l) = \max\{\tilde{f}'(b_l), \tilde{f}'(b_r)\}$ (case 3.2), then $[b_l, b_r]$ is of type (IIb).

Table 4 Combined types of intervals $[b_l, b_r]$

Type	\tilde{f}' and R	\tilde{f}''	squeeze and hat
IIa IVa	$\tilde{f}'(b_l) \geq R \geq \tilde{f}'(b_r)$	$\tilde{f}''(b_l) \leq 0$	$\tilde{r}(x) \leq \tilde{f}(x) \leq \tilde{t}_l(x)$
IIb IVa	$\tilde{f}'(b_l) \geq R \geq \tilde{f}'(b_r)$	$\tilde{f}''(b_r) \leq 0$	$\tilde{r}(x) \leq \tilde{f}(x) \leq \tilde{t}_r(x)$
IIIa IVb	$\tilde{f}'(b_l) \leq R \leq \tilde{f}'(b_r)$	$\tilde{f}''(b_r) \geq 0$	$\tilde{t}_r(x) \leq \tilde{f}(x) \leq \tilde{r}(x)$
IIIb IVb	$\tilde{f}'(b_l) \leq R \leq \tilde{f}'(b_r)$	$\tilde{f}''(b_l) \geq 0$	$\tilde{t}_l(x) \leq \tilde{f}(x) \leq \tilde{r}(x)$

Otherwise, we have $\tilde{f}'(b_l) \geq \tilde{f}'(p) \geq \tilde{f}'(b_r)$ (case 3.3) and by Lemma 2(b) $\tilde{f}''(p) \leq 0$. We then compare $\tilde{f}'(p)$ to the values of the two tangents $\tilde{t}_l(p)$ and $\tilde{t}_r(p)$. Again we have three subcases. If $\tilde{f}'(p) > \tilde{t}_l(p)$ (case 3.3.1), then \tilde{f} cannot be concave in $[b_l, p]$ and thus inflection point p^* is contained in subinterval $[b_l, p]$. In particular we find $\tilde{f}''(b_l) \geq 0 \geq \tilde{f}''(p)$ and $0 \geq \tilde{f}''(b_r)$ which implies that $[b_l, b_r]$ is of type (IIb). Similarly, if $\tilde{f}'(p) > \tilde{t}_r(p)$ (case 3.3.2), then we have $\tilde{f}''(p) \leq 0 \leq \tilde{f}''(b_r)$ and thus $[b_l, b_r]$ is of type (IIa).

Otherwise we have $\tilde{f}'(b_l) \geq R \geq \tilde{f}'(b_r)$, $\tilde{f}'(b_l) \geq \tilde{f}'(p) \geq \tilde{f}'(b_r)$, $\tilde{f}''(p) \leq 0$, and $\tilde{f}'(p) \leq \min\{\tilde{t}_l(p), \tilde{t}_r(p)\}$ (case 3.3.3). In this case we split the given interval into subintervals $[b_l, p]$ and $[p, b_r]$. There is at most one (inflection) point y^* in each subinterval. Hence \tilde{f} is either concave in $[b_l, p]$ (type IVa) or there is an inflection point with $\tilde{f}''(b_l) \geq 0 \geq \tilde{f}''(p)$. As $\tilde{f}'(p) \leq \tilde{t}_l(p)$ we then have $R_l = \frac{\tilde{f}(p) - \tilde{f}(b_l)}{p - b_l} \leq \tilde{f}'(b_l)$. We also have $R_l \geq \tilde{f}'(p)$ since otherwise we had type (Ia) and $\tilde{f}''(b_l) < 0$. Consequently, if there is an inflection point we have type (IIIb). Thus we have type (IVa) or (IIIb). We denote this combined type by (IIIb | IVa). Similarly we find that $[p, b_r]$ is of type (IIIa | IVa).

Case 4 with $\tilde{f}'(b_l) \leq R \leq \tilde{f}'(b_r)$ follows analogously to case 3. \square

Remark 4 Note that as we do not use \tilde{f}'' we can not decide where \tilde{f} is concave or convex in the cases 3.3.3 and 4.3.3 of Table 3. This can only occur for intervals of the starting partition. We therefore always have to split intervals of the starting partition for which \tilde{f} is convex or concave. Nevertheless, it is still possible to use tangents and secants for creating hat and squeeze in such intervals.

3.2 Splitting Intervals

Once we have information about the sign of \tilde{f}'' at b_l and b_r we can derive its sign at possible cutting points when we want to refine the partitioning of the domain. The following proposition allows to calculate the sign of \tilde{f}'' at a cutting point c . The type of the two subintervals then can be determined by means of the derivatives of \tilde{f}' at the new boundary points and the slope R of the secant according to Tables 2 and 4. However, it might be necessary to shift a given cutting point to some point $c_\delta = c + \delta$ for some small $\delta > 0$.

Theorem 3 Let \tilde{f} be a piecewise C^2 -function on domain $[b_l, b_r]$ such that Conditions (C2) and (C3a) hold. Assume that the signs of $\tilde{f}''(b_l)$ or $\tilde{f}''(b_r)$ are known. Then the signs of \tilde{f}'' at points $b_l < c < c_\delta < b_r$ are determined as given by Table 5.

Table 5 Signs of \tilde{f}'' at cutting points c and $c_\delta = c + \delta$ for each type of interval

Type	splitting point
Ia, IIa, IIIa	$\tilde{f}''(b_l) \leq 0 \leq \tilde{f}''(b_r), \quad \tilde{f}'(c) \leq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c_\delta) \geq 0$ $\tilde{f}'(c) \geq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c) \leq 0$
Ib, IIb, IIIb	$\tilde{f}''(b_l) \geq 0 \geq \tilde{f}''(b_r), \quad \tilde{f}'(c) \leq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c) \geq 0$ $\tilde{f}'(c) \geq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c_\delta) \leq 0$
IVa	$\tilde{f}''(b_l), \tilde{f}''(b_r) \leq 0 \Rightarrow \tilde{f}''(c) \leq 0$
IVb	$\tilde{f}''(b_l), \tilde{f}''(b_r) \geq 0 \Rightarrow \tilde{f}''(c) \geq 0$
IIa IVa	$\tilde{f}''(b_l) \leq 0, \quad \tilde{f}'(c) \leq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c_\delta) \geq 0, \quad \tilde{f}''(b_r) \geq 0$ $\tilde{f}'(c) \geq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c) \leq 0$
IIIb IVb	$\tilde{f}''(b_l) \geq 0, \quad \tilde{f}'(c) \leq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c) \geq 0$ $\tilde{f}'(c) \geq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c_\delta) \leq 0, \quad \tilde{f}''(b_r) \leq 0$
IIb IVa	$\tilde{f}''(b_r) \leq 0, \quad \tilde{f}'(c) \leq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c) \geq 0, \quad \tilde{f}''(b_l) \geq 0$ $\tilde{f}'(c) \geq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c_\delta) \leq 0$
IIIa IVb	$\tilde{f}''(b_r) \geq 0, \quad \tilde{f}'(c) \leq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c_\delta) \geq 0$ $\tilde{f}'(c) \geq \tilde{f}'(c_\delta) \Rightarrow \tilde{f}''(c) \leq 0, \quad \tilde{f}''(b_l) \leq 0$

Proof For the proof we restate these implications in Table 6 in a more condensed “raw” form. Table 5 then follows immediately from the characterizations of the corresponding types. Condition “ $\tilde{f}''(b_l) \leq 0$ ” in Table 6 means (in abuse of language) that \tilde{f} is either concave on $[b_l, b_r]$ or there exists a $y \in (b_l, b_r)$ such that \tilde{f} is concave on $[b_l, y^*]$ and convex on $[y^*, b_r]$. Analogously for the other three cases. Because of Condition (C3a) one of these cases applies.

Case (1) - corresponds to types (Ia), (IIa), (IIIa), (IVa), and (IIa | IVa): As \tilde{f}' is concave near b_l there is at most one minimum (or interval of minimums) and no maximum of \tilde{f}' in interior (b_l, b_r) . Now if $\tilde{f}'(c) \leq \tilde{f}'(c_\delta)$, then $\tilde{f}'(c_\delta) \leq \tilde{f}'(b_r)$ and hence $\tilde{f}''(c_\delta) \geq 0$ by Lemma 2(b). Moreover, $\tilde{f}''(b_r) \geq 0$ as \tilde{f}'' cannot change sign in $[c_\delta, b_r]$. Otherwise we have $\tilde{f}'(c) \geq \tilde{f}'(c_\delta)$. Then $\tilde{f}'(b_l) \geq \tilde{f}'(c)$ and hence $\tilde{f}''(c) \leq 0$ by Lemma 2(b).

Cases (2)–(4) follow completely analogously. \square

Remark 5 The exact value of shifting δ is not crucial as we are only interested in the sign of $\tilde{f}''(c)$ in opposition to methods for numerical derivation. Although we

Table 6 Signs of \tilde{f}'' at cutting points c and $c_\delta = c + \delta$

(1)	$\tilde{f}''(b_l) \leq 0,$	$\tilde{f}'(c) \leq \tilde{f}'(c_\delta) \Rightarrow$	$\tilde{f}''(b_l) \leq 0,$	$\tilde{f}''(c_\delta) \geq 0,$	$\tilde{f}''(b_r) \geq 0$
		$\tilde{f}'(c) \geq \tilde{f}'(c_\delta) \Rightarrow$	$\tilde{f}''(b_l) \leq 0,$	$\tilde{f}''(c) \leq 0$	
(2)	$\tilde{f}''(b_l) \geq 0,$	$\tilde{f}'(c) \leq \tilde{f}'(c_\delta) \Rightarrow$	$\tilde{f}''(b_l) \geq 0,$	$\tilde{f}''(c) \geq 0$	
		$\tilde{f}'(c) \geq \tilde{f}'(c_\delta) \Rightarrow$	$\tilde{f}''(b_l) \geq 0,$	$\tilde{f}''(c_\delta) \leq 0,$	$\tilde{f}''(b_r) \leq 0$
(3)	$\tilde{f}''(b_r) \leq 0,$	$\tilde{f}'(c) \leq \tilde{f}'(c_\delta) \Rightarrow$	$\tilde{f}''(b_l) \geq 0,$	$\tilde{f}''(c) \geq 0,$	$\tilde{f}''(b_r) \leq 0$
		$\tilde{f}'(c) \geq \tilde{f}'(c_\delta) \Rightarrow$		$\tilde{f}''(c_\delta) \leq 0,$	$\tilde{f}''(b_r) \leq 0$
(4)	$\tilde{f}''(b_r) \geq 0,$	$\tilde{f}'(c) \leq \tilde{f}'(c_\delta) \Rightarrow$		$\tilde{f}''(c_\delta) \geq 0,$	$\tilde{f}''(b_r) \geq 0$
		$\tilde{f}'(c) \geq \tilde{f}'(c_\delta) \Rightarrow$	$\tilde{f}''(b_l) \leq 0,$	$\tilde{f}''(c) \leq 0,$	$\tilde{f}''(b_r) \geq 0$

want to replace our choice of cutting point c by one which is quite close, δ need not be very small so that we can avoid possible round-off errors. So, e.g., the choice $\delta = |b_r - b_l|/1000$ is fine.

4 The Algorithm

Now we can compile an algorithm that is based on the results of this paper. Algorithm 1 presents Algorithm Tinflex-2 when $c = 0$, i.e., when $T_c(x) = \log(x)$. It is obvious that this algorithm can easily be generalized for arbitrary transformations T_c from Table 1. It is quite straight-forward to compute $T_c(f(x))$ and its derivative from $f(x)$ or $\log(f(x))$ and their corresponding derivatives.

For $c < 0$, however, one must check whether a tangent results in a valid (bounded) hat function. Otherwise, the corresponding interval has to be split. This can be implemented by setting the area in such intervals to $A_{h,i} = \infty$. Although one should also check that \tilde{f} is concave and strictly monotone in the possibly unbounded intervals $(-\infty, b_1]$ and $[b_{n-1}, \infty)$ of the given starting partition, in practice it is only necessary that there is at most one inflection point of \tilde{f} within each of them. It is then quite easy to detect an inflection point in one of these unbounded intervals since then the construction of a hat function fails. In such cases, we set $A_{h,i} = \infty$, and in the next cycle of derandomized adaptive rejection sampling (Steps 7–14), the interval will be split.

An advantage of the proposed algorithm is that the intervals can be treated independently from each other, i.e., we virtually have distinct and possibly different densities within each of the mutually exclusive intervals which make up the domain. The proposed algorithm thus allows (mostly) arbitrary values of c which may differ on different intervals of the starting partition. Another advantage of the proposed algorithm is that it works for any multiple of a density of f . There is thus no need to compute a normalization constant.

Algorithm 1: Algorithm Tinflex-2

Input: Log-density \tilde{f} with domain (b_l, b_r) and its derivative \tilde{f}' together with partition $b_l = b_0 < b_1 < \dots < b_{n-1} < b_n = b_r$ that satisfy Conditions (C1)–(C3); maximal accepted value for ρ_{\max} .
Optional: Types of intervals $[b_i, b_{i+1}]$.

Output: Random variate X with density f .

```

// Setup: Initial intervals
1 for  $i = 0, \dots, n$  do
2   | Compute  $\tilde{f}(b_i)$  and  $\tilde{f}'(b_i)$ .
3 forall intervals  $[b_i, b_{i+1}]$  do
4   | Determine type of interval using Table 3 (if not provided).
5   | Compute intercepts  $\alpha$  and slopes  $\beta$  of hat  $\tilde{h}_i$  and squeeze  $\tilde{s}_i$  using Tables 2 and 4.
6   | Compute area  $A_{h,i}$  below hat and area  $A_{s,i}$  below squeeze using formula (2).

// Setup: Derandomized adaptive rejection sampling
7 repeat
8   |  $A_h \leftarrow \sum A_{h,i}$  and  $A_s \leftarrow \sum A_{s,i}$ .
9   |  $\bar{A} \leftarrow (A_h - A_s) / (\# \text{ intervals})$ .
10  | forall intervals with  $(A_{h,i} - A_{s,i}) > \bar{A}$  do
11  |   | Split interval using ‘‘arc-mean’’ (6).
12  |   | Determine type of interval using Tables 5, 2 and 4.
13  |   | Compute hat, squeeze and areas for the two new intervals.
14 until  $A_h / A_s \leq \rho_{\max}$ .

// Generation
15 loop
16 | Generate  $J$  with probability vector proportional to  $(A_{h,1}, A_{h,2}, \dots)$ .
17 | Generate  $X$  with density prop. to  $h_J$  using formula (3).
18 | Generate  $U \sim U(0, 1)$ .
19 | if  $U h(X) \leq s(X)$  then // evaluate squeeze
20 |   | return  $X$ .
21 | else if  $U h(X) \leq \exp(\tilde{f}(X))$  then // evaluate density
22 |   | return  $X$ .
23 | else
24 |   | Repeat.
```

Also note that Step 16 can be executed in constant time (i.e., independent of the number of intervals) by means of the alias method or the guide table method, see, e.g., [13, Sect. 3].

It is also possible to replace Steps 7–14 (*derandomized* adaptive rejection sampling) by adaptive rejection sampling. However, in opposition to the method proposed by [11] a rejected point is usually not a good choice for splitting an interval. So we suggest to use the point from (6) instead in the interval.

We have coded a proof-of-concept implementation of Algorithm Tinflex-2 and added a ready-to-use version to our **R** package Tinflex [14].

Algorithm `Tinflex-2` is well suited for the fixed parameter setting where we wish to simulate a large number of IID draws from the same density. In particular the marginal generation times hardly depend on the distribution when ρ is (very) close to 1.

A drawback of Algorithm `Tinflex-2` is its time consuming setup step. Thus it may not be the first choice in the dynamic parameter setting where parameters of the distribution are changed from one call of the generator to the next. But it is possible to choose a rather large value of ρ and thus reduce the overhead of the setup part at the expense of increased marginal generation time. In addition one may start with few initial intervals and continue with adaptive rejection sampling. In fact, Gilks and Wild [11] have invented ARS for this dynamic parameter case required for Gibbs sampling, but we have not implemented ARS in our **R** package.

5 Applications

The main advantage of Algorithm `Tinflex-2` is that it can generate from arbitrary distributions with continuous densities. It is only necessary to provide a function that evaluates the log-density and its derivative together with a starting partition into intervals that all include at most one inflection point of the transformed density. We here illustrate the use of `Tinflex-2` applying it to some practical relevant distribution families for which random variate generation is difficult. A first main result is that `Tinflex-2` worked without problem for all distributions we tried. In addition we observed that for $c = -0.5$ the sampling is significantly faster than for $c = 0$.

Botts et al. [6] discuss how `Tinflex` can be used to generate from the Generalized Inverse Gaussian (GIG) distribution. They explain especially how the mode and the minimum of the log-concavity can be used to form the starting partition for the set-up of `Tinflex`. In this contribution we show three more examples. In Sect. 5.1 we apply the method to the Generalized Hyperbolic distribution which is important in financial simulations but has a rather cumbersome density. In Sect. 5.2 we discuss the problem of truncated distributions. And finally we present in Sect. 5.3 an example from spatial statistics.

5.1 Generalized Hyperbolic distribution

The Generalized Hyperbolic (GH) distribution introduced by Barndorff-Nielsen [1] is a very flexible distribution family with semi-heavy tails. It is the mean-variance mixture of a normal and a GIG distribution and popular especially for return modeling in finance. Its density is proportional to

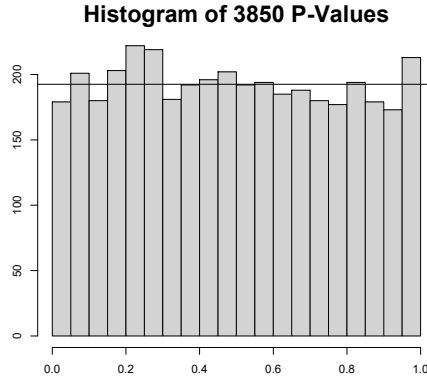


Fig. 2 P-values of Chi-Square tests for 3850 different parameter settings, each with $n = 10^6$

$$f(x) = e^{\beta(x-\mu)} \frac{K_{\lambda-1/2} \left(\alpha \sqrt{\delta^2 + (x-\mu)^2} \right)}{\left(\sqrt{\delta^2 + (x-\mu)^2} / \alpha \right)^{1/2-\lambda}},$$

where $K_\nu(\cdot)$ denotes the modified Bessel function of the third kind. Maybe that complicated density is the reason why there seems to be no rejection algorithm for the GH-distribution available in the literature. Instead most authors suggest to generate GH variates using its mean-variance-mixture representation. It is clear that this generation can not be very fast as a variate of the GIG distribution, a normal variate and a square root are necessary for that approach.

We find it remarkable that our new algorithm `Tinflex-2` can be applied to the GH distribution family to realize a direct generation algorithm. First it is necessary to implement the log-density of the GH distribution and its derivative. This requires the derivative of the modified Bessel function of the third kind, which is equal to $K_\nu(x)' = -K_{\nu-1}(x) - (\nu/x) K_\nu(x)$. It is possible to show that for $c = -0.5$ the transformed density $\tilde{f}(x)$ is concave everywhere or can have on one or both sides of the mode a single interval where $\tilde{f}(x)$ is convex. To find the starting partition without using the second derivative it is easiest to use a numeric search algorithm that finds the minimum of the slope of secants of \tilde{f}' . If that minimum is positive the transformed density \tilde{f} is concave and we can use the mode and an arbitrary point right and left of the mode as starting partition. If the minimum of the slopes of the secants is negative on one or both sides of the mode the points where these minima are attained can be used together with the mode to form the starting partition.

To test the random variates generated with `Tinflex-2` extensively we used $c = -0.5$ and $\rho_{\max} = 1.001$. For 3850 different parameter values we made the chi-square test with sample-size one million. The histogram of those 3850 P-values (see Fig. 2) confirms that the P-values follow the uniform distribution and thus implies that the generated samples follow the correct distribution.

For $c = -0.5$, $\rho = 1.001$ and a sample of size ten million the generation time, including the set-up, is on our standard laptop around 0.36 seconds, compared to 0.5 seconds for the standard **R**-command `rnorm()`. And this speed of **Tinflex-2** for generating from the GH distribution is not influenced by the GH-parameters selected. This is worth to note as it is not the case for GH generation methods based on the GIG generator of Dagpunar [7]. There the running times become extremely slow when $|\lambda| < 0.5$ and parameters α and δ are (very) close to 0. But also for all other parameter settings we have tested the execution time for generating a sample of size ten million is above 1.6 seconds for the **R**-libraries `ghyp`, `fBasics` and `GeneralizedHyperbolic`. We can conclude that using **Tinflex-2** we can generate large samples from the GH distribution at least 4 times faster than using the standard algorithm suggested in the literature.

5.2 Truncated Distributions

One general advantage of transformed density rejection is that it can be used easily for truncated distributions. To generate truncated random variables from the GH distribution we can use directly the approach explained in Section 5.1 above. It is only necessary to find a starting partition of the truncation interval such that each sub-interval holds at most one inflection point of the transformed density. Clearly that can be done using the starting partition P_R that was obtained for the whole real line by first removing from P_R all points that are not in the truncation interval and then adding the lower and upper border of the truncation interval to the remaining points. That means of course that if no point of P_R is in the truncation interval the starting partition for the truncated distribution is only the truncation interval itself. Note that this allows to generate without any problems from the truncated GH distribution with arbitrary truncation intervals. We observed in our experiments that the generation of samples of size ten million from the truncated GH-distribution takes, like for the non-truncated case, approximately 0.36 seconds. Our stable implementation of the log-density of GH and its derivative allows us to generate from truncation intervals in the far tails with extremely small probabilities. For example for $\lambda = .3$, $\alpha = .2$, $\beta = 0.02$, $\delta = .01$ and $\mu = 0$ we generated large samples of the GH-distribution for several different truncation intervals including the interval (1000, 1005). We are not aware of any paper or software that describes the efficient generation of random variates from the GH-distribution truncated to arbitrary intervals.

We also generated large samples from the truncated normal and from the truncated gamma distribution with shape parameter larger than one and different truncation intervals. As expected there occurred no problems and generating ten million variates takes less than 0.35 seconds; again clearly less than generating from the normal distribution (0.5 seconds) or from the gamma distribution (between 0.66 to 1.1 seconds) using the usual **R**-commands. That the efficient generation of truncated standard distributions is practically relevant and not trivial to achieve, can be seen

from two recent papers of Botev and L'Ecuyer [4, 3] implemented in **R** package `TruncatedNormal` [2].

5.3 Watson Distributions

The Watson distribution is used in the modeling of axially symmetric data in spatial statistics. A random unit length vector \mathbf{X} in \mathbb{R}^d has a Watson distribution with concentration parameter $\kappa \in \mathbb{R}$ and mean direction parameter $\boldsymbol{\mu} \in \mathbb{R}^d$ (with $\|\boldsymbol{\mu}\|_2 = 1$) if its density is proportional to $f(\mathbf{x}) \propto \exp(\kappa \boldsymbol{\mu}'\mathbf{x})$. We refer to [16] and the literature cited therein for more details. For sampling from this multivariate distribution we can use the identity that for $\boldsymbol{\mu} = (0, \dots, 0, 1)$, $\mathbf{X} = (\sqrt{1 - W^2}\mathbf{Y}, W)$, where \mathbf{Y} is uniformly distributed on the hypersphere orthogonal to $\boldsymbol{\mu}$ and W has log-density

$$g(w) = \kappa w^2 + \frac{d-3}{2} \log(1 - w^2)$$

on domain $[0, 1]$. It is straightforward to verify that $g(w)$ has at most one inflection point and thus `Tinflex-2` can be applied with its domain as starting partition. Sablica et al. [16] have already shown that the predecessor from [6] can be used for this purpose. However, the new version works as well with about the same marginal running time (which is quite similar to the two examples above) but has a simpler user interface.

6 Conclusions

The algorithm presented in this paper is a user-friendly adaptive acceptance-rejection algorithm. It is user-friendly in the sense that hat and squeeze functions of f are constructed automatically without the user having to know the exact location of the inflection points of the transformed density. The only input required from the user is the transformation T_c (in practice in most cases $c = 0$ or $c = -0.5$), the log-density and its derivative and a partition of the domain of f such that the transformed density does not have more than one inflection point in any of the sub-intervals. The new algorithm improves the method of [6] in the sense that there is no necessity to compute and implement the second derivative of the (log-) density. Our experiments show that the new algorithm is well suited to generate random variates from the Generalized Inverse Gaussian distribution, from the Generalized Hyperbolic distribution and from the Watson distribution. Also random variates from truncated distributions can be generated without problems.

References

1. Barndorff-Nielsen, O.E.: Exponentially decreasing distributions for the logarithm of particle size. *Proceedings of the Royal Society London A* **353**, 401–419 (1977)
2. Botev, Z., Belzile, L.: *TruncatedNormal: Truncated Multivariate Normal and Student Distributions* (2021). URL <https://CRAN.R-project.org/package=TruncatedNormal>. R package version 2.2.2
3. Botev, Z., L'Ecuyer, P.: Simulation from the Normal distribution truncated to an interval in the tail. In: 10th EAI International Conference on Performance Evaluation Methodologies and Tools. ACM (2017). DOI 10.4108/eai.25-10-2016.2266879
4. Botev, Z.I., L'Ecuyer, P.: Efficient probability estimation and simulation of the truncated multivariate Student-t distribution. In: 2015 Winter Simulation Conference (WSC), pp. 380–391 (2015). DOI 10.1109/WSC.2015.7408180
5. Botts, C.: A modified adaptive accept-reject algorithm for univariate densities with bounded support. *Journal of Statistical Computation and Simulation* **81**(3), 1039–1053 (2011)
6. Botts, C., Hörmann, W., Leydold, J.: Transformed density rejection with inflection points. *Statistics and Computing* **23**(2), 251–260 (2012). DOI 10.1007/s11222-011-9306-4
7. Dagpunar, J.S.: An easily implemented generalised inverse Gaussian generator. *Communications in Statistics – Simulation and Computation* **18**(2), 703–710 (1989). DOI 10.1080/03610918908812785
8. Devroye, L.: A simple algorithm for generating random variates with a log-concave density. *Computing* **33**(3–4), 247–257 (1984)
9. Devroye, L.: *Non-Uniform Random Variate Generation*. Springer-Verlag, New-York (1986)
10. Evans, M., Swartz, T.: Random variable generation using concavity properties of transformed densities. *Journal of Computational and Graphical Statistics* **7**(4), 514–528 (1998)
11. Gilks, W.R., Wild, P.: Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* **41**(2), 337–348 (1992)
12. Hörmann, W.: A rejection technique for sampling from T-concave distributions. *ACM Trans. Math. Software* **21**(2), 182–193 (1995)
13. Hörmann, W., Leydold, J., Derflinger, G.: *Automatic Nonuniform Random Variate Generation*. Springer-Verlag, Berlin Heidelberg (2004)
14. Leydold, J., Botts, C., Hörmann, W.: *Tinflex: A Universal Non-Uniform Random Number Generator* (2022). URL <https://CRAN.R-project.org/package=Tinflex>. R package version 2.1
15. Leydold, J., Janka, E., Hörmann, W.: Variants of transformed density rejection and correlation induction. In: K.T. Fang, F.J. Hickernell, H. Niederreiter (eds.) *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pp. 345–356. Springer-Verlag, Heidelberg (2002)
16. Sablica, L., Hornik, K., Leydold, J.: Random sampling from the Watson distribution. *Research Report Series / Department of Statistics and Mathematics 134*, WU Vienna University of Economics and Business, Vienna (2022). URL <https://epub.wu.ac.at/8582/>