

Random Variate Generation for Unimodal and Monotone Densities

L. Devroye, Montreal

Received December 3, 1981; revised May 10, 1982

Abstract — Zusammenfassung

Random Variate Generation for Unimodal and Monotone Densities. We consider the problem of generating random variates with a monotone nonincreasing density on $[0, \infty)$. No bounds are known that would allow a straightforward application of the rejection method, and the inverse of the distribution function is not explicitly known either. We develop the inversion/rejection method, and show how it can be used for all monotone densities, even those with an infinite peak at 0 and unbounded support, provided only that the density f and the distribution function F can be computed for each x . A theoretical analysis of the average time behaviour of the algorithms is included.

AMS Subject Classifications: 65C10, 65C05.

Key words and phrases: Random variate generation, rejection method, inversion method, unimodality, Khintchine's theorem, average time analysis, Newton-Raphson method, table methods.

Erzeugung von Zufallsvariablen mit monotonen oder unimodalen Dichtefunktionen. Wir betrachten das Problem der Erzeugung von Zufallsvariablen mit monoton nichtsteigender Dichtefunktion im Intervall $[0, \infty)$. Schranken, die eine direkte Anwendung der Zurückweisungsmethode erlauben würden sowie die Umkehrfunktion der Verteilung sind nicht bekannt. Wir entwickeln die Umkehr/Zurückweisungsmethode und zeigen ihre Anwendbarkeit auf alle monotonen Dichten, sogar auf solche, die eine Polstelle bei 0 besitzen und die einen unbeschränkten Wertebereich haben. Vorausgesetzt ist lediglich, daß f und F an jeder Stelle berechenbar sind. Eine theoretische Analyse des mittleren Zeitverhaltens der Algorithmen ist beigefügt.

1. Introduction

In this paper we give algorithms that can be used for the computer generation of random variables with a unimodal density f when no bounds are available for f that would allow us to use the rejection method in a straightforward manner. We assume that a perfect uniform $[0, 1]$ random variate generator is given, capable of generating an i. i. d. (independent identically distributed) sequence U_1, U_2, \dots of uniform $[0, 1]$ random variates. The techniques considered here are

- (i) *general*: they can be applied to all unimodal densities with given mode, regardless of the size of the tail or the height of the peak; the densities are not assumed to belong to a parametric family, nor are bounds needed for them;

- (ii) *exact*: if all the operations can be carried out with infinite precision, then the generated random variates have density f ; no approximations are allowed;
- (iii) *efficient*: the average time needed per random variable should be reasonable. Of course, general algorithms of the type given here cannot be expected to be faster than algorithms designed for specific families of densities.

We will try, wherever possible, to give the statistical properties of the algorithms such as the average number of iterations per random variate, and so forth. For example, if T is the time taken by an algorithm, then we are looking for simple expressions and simple upper bounds for $E(T)$, the average time per random variate. These should depend upon general constants only, such as the supremum of f , the support of f , or parameters used in the design of the algorithm. General algorithms with well-understood properties will survive longer than ad hoc algorithms that are known to perform well on specific examples but whose general properties are not clear. Our results about $E(T)$ are based upon the not so unrealistic assumption that the common operations $+$, $-$, $*$, $/$, mod , truncate, compare, move, generate a uniform random variate, \log , and \exp take a given constant time.

The reader is assumed to have a basic knowledge of the common principles in non-uniform random variate generation, such as the principles of inversion, rejection, squeezing, composition, and aliasing (see Schmeiser (1980) for a survey with bibliography). The present paper is a short version of a survey and study reported in Devroye (1982).

2. The Inversion/Rejection Method

There are situations in which the rejection method can be applied with minimal knowledge about f , e.g. it can be applied whenever we know that f is bounded by c , and the support of f is contained in $[0, 1]$. But if we are not given the constant c in this example, it is not clear how one should proceed without a drastic modification of the rejection method.

If F is the distribution function of f , then $F^{-1}(U)$ has density f when U is uniformly distributed on $[0, 1]$. Unfortunately, we are usually not given F^{-1} explicitly, so that the inversion method must be implemented through the numerical solution of the equation $F(X) = U$ for X . This would, strictly speaking, take an infinite amount of time. The main contribution of this paper is the development of the inversion/rejection method applicable when both f and F are given (but not F^{-1}), and when f is known to belong to a broad class of densities (such as all unimodal densities).

The principle is simple: partition R into a countable number of intervals A_1, A_2, \dots (this is a fixed partition), and let p_i be the probability under f of A_i (this is computable since F is given). Then, proceed as follows:

Step 1: [*Inversion.*] Generate a uniform $[0, 1]$ random variate U , set $I \leftarrow 1$, $S \leftarrow p_I$.
 While $U > S$ do: $I \leftarrow I + 1$, $S \leftarrow S + p_I$.
 [I now satisfies $P(I = i) = p_i$, all i . It could have been generated by a method other than sequential search, but that would require the storage of the p_i 's.]

Step 2: [Rejection.] Exit with a random variate X with density f restricted to A_i . Employ the rejection method by deriving a bound for f based upon the knowledge about f (unimodality, etc.).

To illustrate this on a trivial example, assume that we know that $f=0$ on $(-\infty, 0)$ and that $f \leq c$ on $[0, \infty)$. Nothing else is known about f . If we take $A_i = \left[\frac{i-1}{c}, \frac{i}{c} \right)$, $i=1, 2, \dots$, then the algorithm reads:

Step 1: Generate a uniform $[0, 1]$ random variate U . Set $I \leftarrow 1$, $S \leftarrow F\left(\frac{1}{c}\right)$.

While $U > S$ do: $I \leftarrow I + 1$, $S \leftarrow F\left(\frac{I}{c}\right)$.

Step 2: Generate V, W , two independent uniform $[0, 1]$ random variates.

Set $X \leftarrow \frac{I-1+W}{c}$.

If $Vc < f(X)$, exit with X . Otherwise, go to 2.

We do not claim that this is a fast algorithm: in fact, the average time per random variate is ∞ ! The fundamental reason behind the inherent slowness is the size of the family of densities. By appropriately limiting the class of densities, faster algorithms can be constructed.

We will limit ourselves to the class of *monotone densities* on $[0, \infty)$, i.e. $f=0$ on $(-\infty, 0)$, f is nonincreasing on $[0, \infty)$. We will allow $\lim_{x \downarrow 0} f(x) = \infty$. When a density is *unimodal* with mode at m , it can be cut into a monotone density on $[m, \infty)$, and a symmetrically defined monotone density on $(-\infty, m]$, with weights $F(m)$ and $1-F(m)$ to be given to each part. Thus, by the composition method, random variates with a unimodal density with mode at m can be generated if we know how to generate random variates with a monotone density on $[0, \infty)$. We note here that this argument does not apply when the mode is unknown.

In the next section, we will give a brief survey of general algorithms that can be used for generating random variates with a monotone density. Some of these algorithms require some additional knowledge, such as the support of f (or an interval containing the support of f), the supremum of f (which is $B=f(0)$, but $f(0)$ could be ∞), or the knowledge that f is concave or convex on its support.

3. Algorithms

3.1. Inversion

For monotone densities, F is concave on $[0, \infty)$. Thus, the solution of $F(X)=U$ (where U is uniformly distributed on $[0, 1]$) can be obtained by the Newton-Raphson method, and convergence is guaranteed in all cases:

Step 1: Generate a uniform $[0, 1]$ random variate U .

If f is not known to be bounded: $X \leftarrow 2$; repeat $X \leftarrow X/2$ until $F(X) < U$.

If f is bounded: $X \leftarrow 0$.

Step 2: $X \leftarrow X - (F(X) - U)/f(X)$. Go to 2.

This procedure does not halt. In practice, on a finite wordsize computer, one keeps on iterating until the value of X remains unchanged. Thus, the average times obtained in a timing experiment will depend very heavily on the wordsize of the computer, and the distribution function F .

When F is not given, the inversion method is hard to implement. The most valid attempt at obtaining a general algorithm for generating a random variate when only f is given, is that of Ahrens and Kohrt (1981) (see also Kohrt (1980)), based upon the method of guide tables (Chen and Asau (1973)): strictly speaking however, the inversion algorithm given above stopped after a finite number of iterations, and the general method of Ahrens and Kohrt are not exact in the sense (ii) described in the introduction.

3.2. Rejection

In this section we assume only that f is monotone, and that

- (i) $f(x) = 0$ outside $[0, 1]$;
- (ii) $f(x) \leq B$ for all x , where B is known.

If B is not known, it can be computed as $f(0)$, and if the support of f is not known, or exceeds $[0, 1]$, then with a scaling adjustment the support can be made exactly equal to $[0, 1]$ or it can be made equal to $[0, c]$ where c is guaranteed not to exceed 1. The *ordinary rejection method* for generating a random variate X with density f , proceeds as follows:

Step 1: Generate two independent uniform $[0, 1]$ random variates X and V .

Step 2: If $VB \leq f(X)$, exit with X . Otherwise, go to 1.

On the average, step 1 is executed B times. In families of densities with shape parameters, B often depends upon the shape parameter, and can grow very large for some values of the shape parameter. Thus, the ordinary rejection method can be intolerably slow, and we cannot give any guarantees about its speed. If it were not for the fact that this method can be improved upon dramatically without much effort, we would not have mentioned it in this paper.

Because the area under f is 1, we must have

$$f(x) \leq \frac{1}{x}, \quad x > 0,$$

for all monotone densities f . If f is also known to be convex, this bound can be sharpened to $f(x) \leq 1/(2x)$: find the point x where $f'(x) = -1$. The area of the triangle formed by the axes and the tangent to f at x is at most 1 (because f is convex); but the area is exactly $2xf(x)$. To treat both cases simultaneously, we assume that $xf(x) \leq \frac{1}{k}$ where $k = 2, 1$ according to whether f is known to be convex or not.

Thus, we have

$$f(x) \leq \min \left(B, \frac{1}{kx} \right), \quad 0 \leq x \leq 1.$$

The area under the top curve is $\frac{1}{k} (1 + \log(kB))$. It is proportional to the density $g(x) = \min(B, 1/(kx)) k/(1 + \log(kB))$, which has distribution function

$$\begin{cases} \frac{kBx}{1 + \log(kB)}, & 0 \leq x \leq 1/(kB), \\ \frac{1 + \log(xkB)}{1 + \log(kB)}, & 1/(kB) \leq x \leq 1. \end{cases}$$

A random variate with density g can easily be obtained by inversion, and the *modified rejection algorithm* is:

Step 1: Generate two independent uniform $[0, 1]$ random variates U and V .

Step 2: If $U \leq 1/(1 + \log(kB))$, set $X \leftarrow U(1 + \log(kB))/(kB)$. If $VB \leq f(X)$, exit with X . Otherwise, go to 1.

If $U > 1/(1 + \log(kB))$, set $X \leftarrow (kB)^{-1} \exp(U(1 + \log(kB)) - 1)$. If $V \leq kXf(X)$, exit with X . Otherwise, go to 1.

The average number of executions of step 1 is $\frac{1}{k} (1 + \log(kB))$. When $k=1$, this is less than B whenever $B > e$. In most applications the improvement over the ordinary rejection method is noticeable if not spectacular. The computation of kB and $1 + \log(kB)$ should be done in a set-up step. When f is convex ($k=2$), the average number of executions of step 1 is about half of what it was for $k=1$. In a sense, the knowledge that f is convex contributes at least 50% to the useful knowledge about f for random variate generation.

None of the rejection methods have times that remain bounded as $B \rightarrow \infty$, but they are very short and easy to understand; no tables or large set-up times are required either.

3.3. Inversion/Rejection by Halving

In this section we will apply the general inversion/rejection principle to the family of all *monotone* densities f with *support contained in* $[0, 1]$. We assume that f and F can be computed exactly in constant time. The sequential search for the inversion step proceeds by looking at the intervals $[t, 1]$, $[rt, t]$, $[r^2t, rt]$, etc. where $r, t \in (0, 1)$ are constants to be determined. The term “halving” is used because of the obvious popularity of the choice $r = t = \frac{1}{2}$. The algorithm can be reduced to the following form:

Step 1: Generate a uniform $[0, 1]$ random variate U . Set $(X, X^*) \leftarrow (t, 1)$.

While $U > F(X) : (X, X^*) \leftarrow (rX, X)$.

[At the end of this, we know that the solution of $F(x) = U$ belongs to (X, X^*) .]

Step 2: Generate two independent uniform $[0, 1]$ random variates V and W .

Set $Y \leftarrow X + V(X^* - X)$. [Y is uniformly distributed on $[X, X^*]$.]

If $W \leq f(Y)/f(X)$, exit with Y . Otherwise, go to 2.

If f is known to be convex on $[0, 1]$, then we could consider rejection from a trapezoidal dominating curve joining $(X, f(X))$ and $(X^*, f(X^*))$. The changes needed in the algorithm:

In step 1: Compute $Z \leftarrow f(X)$ and $Z^* \leftarrow f(X^*)$ at end of step 1.

In step 2: Replace all of step 2 by the following: generate three independent uniform $[0, 1]$ random variates, U, V and W .

Let $R \leftarrow \min\left(U, V \frac{Z + Z^*}{Z - Z^*}\right)$, $Y \leftarrow X + R(X^* - X)$ (Y has a density proportional to the trapezoid determined on $[X, X^*]$ by the points $[Z, Z^*]$).
Let $T \leftarrow W(Z + R(Z^* - Z))$.

[Squeeze step. Optional.] If $T \leq Z^*$, exit with Y .

[Acceptance/rejection step.] If $T \leq f(Y)$, exit with Y . Otherwise, go to 2.

Remark 1:

The efficiency of the algorithm can be increased by storing a table of constants $(f(x), F(x))$ for $x = 1, t, rt, r^2 t, r^3 t, \dots$. This will pay off when many random variates are needed from the same distribution. Because only finite tables can be stored, we would still need some version of the halving algorithm for the lower part of the interval $[0, 1]$. Also, the statistical properties of the algorithm do not depend upon the presence of a partial table.

There are two big contributors to the average time $E(T)$ of the inversion/rejection algorithm:

- (i) $E(N_s)$: the average number of steps in the sequential interval search;
- (ii) $E(N_r)$: the average number of iterations in the rejection step;
this is equal to the total area under the dominating curve;
in the case of inversion/rejection by halving, the dominating curve is a collection of rectangles with bases $[t, 1]$, $[rt, t]$, etc. and heights $f(t)$, $f(rt)$, etc.

Theorem 1: Let $H(f) = \int_0^1 \log \frac{1}{x} \cdot f(x) dx$. Then

$$\begin{aligned} 1 + \frac{\log t}{\log \frac{1}{r}} + \frac{H(f)}{\log \frac{1}{r}} &\leq 1 + \int_0^t f(x) \cdot \log \frac{t}{x} dx / \log \frac{1}{r} \leq E(N_s) \\ &\leq 1 + \int_0^t f + \int_0^t f(x) \cdot \log \frac{t}{x} dx / \log \frac{1}{r} \leq 2 + \frac{\log t}{\log \frac{1}{r}} + \frac{H(f)}{\log \frac{1}{r}}. \end{aligned}$$

For $t=r$, the simple outer bounds for $E(N_s)$ just read

$$\frac{H(f)}{\log \frac{1}{r}} \leq E(N_s) \leq 1 + \frac{H(f)}{\log \frac{1}{r}}.$$

Also,

$$1 \leq E(N_r) \leq \frac{1}{r} \int_0^{tr} f + f(t)(1-t) \leq \frac{1}{r} \int_0^r f \leq \frac{1}{r},$$

where the last two inequalities only hold when $t=r$.

Proof: Note that

$$E(N_s) = \int_t^1 f + \sum_{i=1}^{\infty} (i+1) \int_{tr^i}^{tr^{i-1}} f = 1 + \sum_{i=1}^{\infty} i \int_{tr^i}^{tr^{i-1}} f.$$

For $x \in [tr^i, tr^{i-1})$, we have $0 \leq i - \log \frac{t}{x} / \log \frac{1}{r} \leq 1$. Thus,

$$0 \leq E(N_s) - \int_0^t f(x) \log \frac{t}{x} \log^{-1} \frac{1}{r} dx - 1 \leq \int_0^t f.$$

All the statements involving $E(N_s)$ follow trivially.

Next, we have

$$\begin{aligned} E(N_r) &= \sum_{i=1}^{\infty} f(tr^i)(tr^{i-1} - tr^i) + f(t)(1-t) \\ &\leq \sum_{i=1}^{\infty} \int_{tr^{i+1}}^{tr^i} f \cdot \frac{tr^{i-1} - tr^i}{tr^i - tr^{i+1}} + f(t)(1-t) \\ &= \frac{1}{r} \int_0^{tr} f + f(t)(1-t) \\ &\leq \frac{1}{r} \int_0^r f \text{ when } t=r. \end{aligned}$$

Theorem 1 shows that regardless of the choice of t and r , the difficulty inherent in f is appropriately measured by $H(f)$. This quantity can of course be infinite, in which case $E(N_s) = \infty$. Often $H(f)$ can be computed or estimated beforehand. In theorem 2 below, we give some inequalities that link $H(f)$ to better known quantities such as the mean, $\sup f$, and the $L \log_+ L$ norm of f .

Theorem 2:

(i) For all monotone densities,

$$H(f) \geq \log \frac{1}{\int x f(x) dx}.$$

(ii) For all monotone densities on $[0, 1]$,

$$1 \leq H(f) \leq 1 + \log(f(0)).$$

(iii) For all monotone densities,

$$\int f \log_+ f \leq H(f) \leq 2 \int f \log_+ f + \frac{4}{e}.$$

Thus, $H(f)$ is finite if and only if $f \log_+ f$ is integrable, i.e. $f \in L \log_+ L$.

Proof: (i) follows from the convexity of $-\log(x)$ and Jensen's inequality (see for example Feller, 1971). The second inequality ($1 \leq H(f)$) uses the fact that $-\log(x)$ and $f(x)$ are both nonincreasing on $[0, 1]$, and therefore, by Steffensen's inequality (1925),

$$\int_0^1 -\log(x) f(x) dx \geq \int_0^1 -\log(x) dx \int_0^1 f(x) dx = 1.$$

The upper bound in (ii) is a special case of another inequality of Steffensen (1918): if $0 \leq f \leq 1$, and if g is nonincreasing and integrable on $[0, 1]$, then

$$\int_0^1 g(x) f(x) dx \leq \int_0^a g(x) dx \quad \text{where} \quad a = \int_0^1 f(x) dx.$$

Let $g(x) = -\log(x)$, and replace $f(x)$ by $f(x)/\sup f$. Thus, $a = 1/\sup f$, and

$$\int_0^a -\log(x) dx = a \left(1 + \log \frac{1}{a} \right)$$

give the desired result. The upper bound in (iii) is a Young-type inequality found in Hardy, Littlewood and Polya (1952, theorem 239). (This inequality does not use the monotonicity of f .) The lower bound in (iii) follows from $f(x) \leq \frac{1}{x}$.

Example 1:

We consider the beta($1, a + 1$) density $f(x) = (a + 1) (1 - x)^a$, $0 \leq x \leq 1$, where $a > 0$ is a parameter. We have $\sup f = a + 1$, and $\int x f(x) dx = 1/(a + 2)$. By (i) and (ii) of theorem 2, $\log(a + 2) \leq H(f) \leq 1 + \log(a + 1)$, and thus $H(f) \sim \log a$ as $a \rightarrow \infty$. By theorem 1, we can then conclude that if r, t are fixed, $E(T)$ must grow as $\log a$ as $a \rightarrow \infty$.

Remark 2: [Convex densities.]

When the modification for convex densities is implemented, the statistical properties of $E(N_r)$ change somewhat, but those of $E(N_s)$ remain the same. For example, it is easy to show that the inequality $E(N_r) \leq \frac{1}{r}$ valid for $t = r$ can be replaced by the tighter inequality $E(N_r) \leq \frac{1}{2} \left(1 + \frac{1}{r} \right)$.

Remark 3: [Choice of r .]

Assume that $t = r$. If we keep r fixed, then $E(N_s) \geq 1 + H(f)/\log \frac{1}{r}$ implies that the average time of the algorithm grows at least as a constant times $H(f)$. Considering

the upper and lower bounds of theorem 1, we have that

$$E(N_s) + E(N_r) \sim H(f) / \log \frac{1}{r} \text{ as } H(f) \rightarrow \infty. \tag{1}$$

But the best r is the one which minimizes $E(N_s) + E(N_r)$, in first approximation (N_r and N_s are given equal weight because they correspond to the number of computations of f and F respectively). A reasonable upper bound for $E(N_r) + E(N_s)$ is $1 + \frac{1}{r} + H(f) / \log \frac{1}{r}$. This is minimal when $\frac{1}{r} \log^2 \frac{1}{r} = H(f)$ (because $H(f) \geq 1$, a solution $r < 1$ always exists). One Newton-Raphson iteration for solving this started at $\frac{1}{r} = H(f)$ gives

$$\frac{1}{r} = \frac{H(f)}{\log(H(f))} \cdot \frac{1 + 2 \log(H(f))}{2 + \log(H(f))} \sim 2 \frac{H(f)}{\log(H(f))} \text{ as } H(f) \rightarrow \infty.$$

If we take this value for $\frac{1}{r}$, or its asymptotic equivalent $2H(f) / \log(H(f))$, then

$$E(N_s) + E(N_r) \leq (1 + 3H(f) / \log(H(f))) (1 + o(1)) \text{ as } H(f) \rightarrow \infty. \tag{2}$$

The asymptotic rate (with respect to $H(f)$) in (2) is better than in (1). Thus, for large values of $H(f)$, it pays to choose r as a function of f . Since $H(f)$ is unknown, this improvement is not implementable. One possible practical solution is outlined in remark 4.

Remark 4: [Choice of r when f is bounded.]

For bounded f , we have $H(f) \leq 1 + \log(f(0))$. Thus, instead of taking the asymptotically optimal value $\frac{1}{r} = 2H(f) / \log(H(f))$, we could consider

$$\frac{1}{r} = 2 \frac{1 + \log(f(0))}{\log(1 + \log(f(0)))}. \tag{3}$$

A little work then shows that

$$E(N_s) + E(N_r) = O\left(\frac{\log(f(0))}{\log(\log(f(0)))}\right) \text{ as } f(0) \rightarrow \infty.$$

For the modified rejection algorithm, we had seen that the average time increased as $\log(f(0))$ as $f(0) \rightarrow \infty$, and this is thus worse than the performance of the inversion/rejection method with halving when r is chosen as in (3). However, for the improvement to show in experimental results, $f(0)$ must be very large (the improvement is of the order of $\log(\log(f(0)))$). In most experiments, the modified rejection algorithm was faster (see section 4 below).

3.4. Inversion/Rejection by Doubling

Assume that f is a monotone density bounded by B , and that f and F can be computed: f can have unbounded support. We organize the interval search by looking at $[0, t)$, $[t, tr)$, $[tr, tr^2)$, ... where $t > 0$ and $r > 1$ are constants. The nickname "doubling" is given here for the obviously convenient choice $r=2$. The inversion/rejection algorithm with doubling can be summarized as follows:

Step 1: Generate a uniform $[0, 1]$ random variate U . Set $X \leftarrow 0$, $X^* \leftarrow t$.
If $U < F(X^*)$, go to 2 (the solution of $F(x) = U$ belongs to $[X, X^*)$).
Otherwise, $X \leftarrow X^*$, $X^* \leftarrow rX^*$, go to 1.

Step 2: Generate two independent uniform $[0, 1]$ random variates V, W .
Set $Y \leftarrow X + (X^* - X)V$. (Y is uniformly distributed on $[X, X^*)$.)
If $W \leq f(Y)/f(X)$, exit with Y . Otherwise, go to 2.

(Version of the algorithm when f is convex on $(0, \infty)$.)

Step 1: Same as above. At time of exit of step 1, set $Z \leftarrow f(X)$, $Z^* \leftarrow f(X^*)$.

Step 2: Generate three independent uniform $[0, 1]$ random variates U, V, W .
Let $R \leftarrow \min\left(U, V \frac{Z + Z^*}{Z - Z^*}\right)$, $Y \leftarrow X + (X^* - X)R$. (Y has a density that is proportional to the trapezoid determined by (X, Z) , (X^*, Z^*) .)
Let $T \leftarrow W(Z + (Z^* - Z)R)$.

Step 3: (Squeeze step. Optional.) If $T \leq Z^*$, exit with Y .
(Acceptance/rejection step.) If $T \leq f(Y)$, exit with Y . Otherwise, go to 2.

Theorem 3: Let f be a monotone density bounded by B , let $t > 0$ and $r > 1$ be constants. If

$$H_t(f) = \int_0^\infty \log_+ \left(\frac{x}{t} \right) f(x) dx,$$

then

$$1 + H_t(f)/\log r \leq E(N_s) \leq 2 + H_t(f)/\log r$$

and

$$1 \leq Bt + \int_t^\infty f(x) dx \leq E(N_r) \leq Bt + r$$

for the inversion/rejection algorithm with doubling. For the version used when f is convex, the last inequality should be replaced by $1 \leq E(N_r) \leq \frac{1}{2}(Bt + r + 1)$.

Proof: We will repeatedly use the fact that $tr^{i-1} \leq x < tr^i$ if and only if

$$i - 1 \leq \log(x/t)/\log r < i, \quad i \geq 1.$$

Now,

$$\begin{aligned} E(N_s) &= \int_0^t f(x) dx + \sum_{i=1}^{\infty} (i+1) \int_{tr^{i-1}}^{tr^i} f(x) dx = 1 + \sum_{i=1}^{\infty} i \int_{tr^{i-1}}^{tr^i} f(x) dx \\ &\leq 2 + \int_t^\infty \frac{\log(x/t)}{\log r} f(x) dx = 2 + H_t(f)/\log r \end{aligned}$$

and

$$E(N_s) \geq 1 + \int_t^\infty \frac{\log(x/t)}{\log r} f(x) dx = 1 + H_t(f)/\log r.$$

Also,

$$\begin{aligned} E(N_r) &= Bt + \sum_{i=1}^\infty (tr^i - tr^{i-1}) f(tr^{i-1}) \\ &\leq Bt + \sum_{i=1}^\infty (tr^i - tr^{i-1})(tr^{i-1} - tr^{i-2})^{-1} \int_{tr^{i-2}}^{tr^{i-1}} f(x) dx \\ &\leq Bt + r, \end{aligned}$$

and

$$\begin{aligned} E(N_r) &\geq Bt + \sum_{i=1}^\infty (tr^i - tr^{i-1})(tr^i - tr^{i-1})^{-1} \int_{tr^{i-1}}^{tr^i} f(x) dx \\ &= Bt + \int_t^\infty f(x) dx \geq 1. \end{aligned}$$

For the convex version, $E(N_s)$ remains the same, and $E(N_r)$ changes slightly:

$$E(N_r) = t \left(\frac{B+f(t)}{2} \right) + \sum_{i=1}^\infty (tr^i - tr^{i-1}) \left(\frac{f(tr^{i-1}) + f(tr^i)}{2} \right)$$

which is equal to the expression of $E(N_r)$ for the non-convex version (take the term with B and the terms with $f(tr^{i-1})$ only) plus an expression not exceeding 1 (take all the other terms) divided by 2. Thus,

$$E(N_r) \leq \frac{1}{2} (Bt + r + 1).$$

This concludes the proof of theorem 3.

We would like this algorithm to perform at a speed that is independent of the scaling of the x -axis. This can be achieved if we set $t = 1/B$. With this choice for t ,

$$E(N_s) + E(N_r) \leq 3 + r + H_{1/B}(f)/\log r. \tag{4}$$

The upper bound does not depend upon the scale, because $H_{1/B}(f)$ depends only upon the shape of f , so we could call it $H^*(f)$. In a sense, $H^*(f)$ measures the difficulty f gives us in random variate generation with the inversion/rejection method based on doubling, and it is the counterpart of $H(f)$ in the halving algorithm. There are of course some densities for which $H^*(f) = \infty$: for example,

$$f(x) = \frac{1}{(x+e)\log^2(x+e)}, \quad x > 0,$$

is monotone, and $B = 1/e$. Such densities cause special problems in simulations because the average number of bits in the integer part of random variates with densities having $H^*(f) = \infty$ is infinite!

In practice, $H^*(f)$ is not known, but other quantities such as $E(X)$ or $E(X^2)$ sometimes are. A loose upper bound for $H^*(f)$ is afforded by Jensen's inequality:

$$H^*(f) \leq \int_0^{\infty} \log(1+Bx) f(x) dx \leq \log(1+E(BX)).$$

Thus, at worst, the average time of the algorithm is logarithmic in $E(BX)$.

Example 1:

We continue example 1 of section 3.3, and note that

$$H^*(f) \leq \log(1+E(BX)) = \log\left(1 + \frac{a+1}{a+2}\right) \leq \log 2, \text{ all } a > 0.$$

Thus,

$$E(N_s) \leq 2 + \frac{\log 2}{\log r}, \quad E(N_r) \leq 1 + r.$$

The "ad hoc" choice $r=2$ makes *both* upper bounds equal to 3. The average time taken in the algorithm is uniformly bounded in a .

Remark 5: [Choice of r .]

The obvious choice $r=2$ gives $E(N_s) + E(N_r) \leq 5 + H^*(f)/\log 2$. The upper bound (4) for $E(N_s) + E(N_r)$ is minimal for the unique value $r > 1$ for which $r \log^2 r = H^*(f)$. Copying the discussion of remark 3, we see that the choice $r = 2 H^*(f)/\log(H^*(f))$ makes the upper bound asymptotic to $3 H^*(f)/\log(H^*(f))$ as $H^*(f) \rightarrow \infty$. Again, this asymptotic rate is better than the one obtained by keeping r fixed: indeed, for r fixed, we have $E(N_s) + E(N_r) \geq 2 + H^*(f)/\log r$. Since $H^*(f)$ is unknown in general, we suggest either estimating it before random variate generation is started (but this would only be feasible if many random variates are needed), or using the value $r=2$.

The inequality $H^*(f) \leq \log(1+E(BX))$ could help in finding a good value for r when $E(BX)$ is known. For example, by taking

$$r = 2 \log(1+E(BX))/\log \log(e+E(BX))$$

we obtain

$$E(N_s) + E(N_r) \leq (3 + o(1)) \log(E(BX))/\log \log(E(BX)) \text{ as } E(BX) \rightarrow \infty.$$

Such a choice of r should be close to the optimal value when the bound $H^*(f) \leq \log(1+E(BX))$ is tight, i.e. the difference between right and left is small.

3.5. Inversion/Rejection via Newton-Raphson Iterations

In this section, we assume, as in section 3.4, that f is monotone and bounded by $B=f(0) < \infty$: f need not have compact support, but both f and F should be computable. The interval search considers the intervals $[x_0, x_1]$, $[x_1, x_2]$, etc. where each x_{n+1} is a function of x_n only (this avoids the problem of having to store the x_n 's), and is computed by the rule

$$x_{n+1} = x_n + (1 - F(x_n))/f(x_n), \quad x_0 = 0. \tag{5}$$

The sequence $\{x_n\}$ thus obtained coincides with the sequence of values obtained if we try to solve the equation $F(x)=1$ for x by Newton-Raphson iterations started at $x_0=0$. Since F is concave, we know that x_n increases monotonically to a finite solution if it exists, and to ∞ if no finite solution exists (i.e., f has no compact support). Thus, for $0 < u < 1$, the solution of $F(x)=u$ certainly belongs to one of the intervals $[x_n, x_{n+1})$.

The advantages of this type of interval search are triple: the average time of the algorithm is scale-invariant; there are no design parameters as for the halving and doubling algorithms; and there is a natural balance between the inversion and rejection steps in the algorithm: $E(N_s)=E(N_r)$ (see theorem 4 below).

Step 1: Generate a uniform $[0, 1]$ random variate U , set $X \leftarrow 0$.
 Compute $R \leftarrow F(X)$, $Z \leftarrow f(X)$.

Step 2: Set $X^* \leftarrow X + (1 - R)/Z$, $R^* \leftarrow F(X^*)$, $Z^* \leftarrow f(X^*)$.
 If $R^* > U$, go to 3. (The solution of $F(x)=U$ belongs to $[X, X^*)$.
 Otherwise, $R \leftarrow R^*$, $Z \leftarrow Z^*$, $X \leftarrow X^*$, go to 2.

Step 3: Generate two independent uniform $[0, 1]$ random variates V, W .
 Set $Y \leftarrow X + (X^* - X)V$, $T \leftarrow WZ$. (Y is uniformly distributed on $[X, X^*)$.)

Step 4: (Squeeze step. Optional.) If $T \leq Z^*$, exit with Y .
 (Acceptance/rejection step.) If $T \leq f(Y)$, exit with Y . Otherwise, go to 3.

Theorem 4 describes a remarkable coincidence: $E(N_s)=E(N_r)$ for all bounded monotone densities. A perfect balance is obtained between the two parts of the algorithm despite the fact that search and rejection are seemingly independent and totally different processes. In other words, on the average, equal amounts of energy are required for the interval search and for the rejection step.

Theorem 4: *Let f be a bounded monotone density, and let $0 = x_0 \leq x_1 \leq \dots$ be the sequence obtained by Newton-Raphson iteration (5). Then the inversion/rejection method with Newton-Raphson iterations satisfies*

$$E(N_s) = E(N_r) = \sum_{i=0}^{\infty} (1 - F(x_i)).$$

Proof:

$$E(N_s) = \sum_{i=1}^{\infty} i((1 - F(x_{i-1})) - (1 - F(x_i))) = \sum_{i=0}^{\infty} (1 - F(x_i))$$

and

$$E(N_r) = \sum_{i=0}^{\infty} f(x_i)(x_{i+1} - x_i) = \sum_{i=0}^{\infty} (1 - F(x_i))$$

by (5). This concludes the proof of theorem 4.

Formula (5) can be rewritten as

$$x_{n+1} = x_n + 1/h(x_n), \quad x_0 = 0,$$

where $h(x) = \frac{f(x)}{1-F(x)}$ is the *hazard rate* of the distribution. The present algorithm performs extremely well for densities with a nondecreasing hazard rate, as confirmed by the experiments of section 4. We will show that the average time is uniformly bounded over the class of all densities with nondecreasing hazard rate. When the hazard rate is nonincreasing, we will see that the average time increases at worst linearly with $E(BX)$, a scale-invariant quantity.

Theorem 5: *Let f be a monotone density bounded by B , and let a random variate X with density f be generated by the inversion/rejection method with Newton-Raphson iterations. Then, if the hazard rate h is nonincreasing, $E(BX) \geq 1$ and*

$$1 \leq E(N_r) = E(N_s) \leq 1 + E(BX).$$

If the hazard rate is nondecreasing, $E(BX) \leq 1$ and

$$1 \leq E(N_r) = E(N_s) \leq \frac{e}{e-1}.$$

Proof: If h is nonincreasing,

$$E(BX) = B \int_0^{\infty} (1-F(x)) dx = \int_0^{\infty} \frac{B}{h(x)} f(x) dx \geq 1.$$

For nondecreasing h , the inequality should be reversed. If h is nonincreasing, we also have

$$\begin{aligned} \sum_{i=0}^{\infty} (1-F(x_i)) &\leq 1 + \sum_{i=1}^{\infty} \int_{x_{i-1}}^{x_i} (1-F(x)) dx / (x_i - x_{i-1}) \\ &= 1 + \sum_{i=1}^{\infty} \int_{x_{i-1}}^{x_i} (1-F(x)) dx h(x_{i-1}) \\ &\leq 1 + \int_0^{\infty} B(1-F(x)) dx = 1 + E(BX). \end{aligned}$$

Finally, when h is nondecreasing, we note that

$$\begin{aligned} (1-F(x_{i+1})) &= (1-F(x_i)) \exp\left(-\int_{x_i}^{x_{i+1}} h(x) dx\right) \\ &\leq (1-F(x_i)) \exp(-h(x_i)(x_{i+1} - x_i)) = (1-F(x_i))/e. \end{aligned}$$

Thus,

$$\sum_{i=0}^{\infty} (1-F(x_i)) \leq \sum_{i=0}^{\infty} e^{-i} = \frac{e}{e-1}.$$

This concludes the proof of theorem 5.

Example 2: [Monotone hazard rates.]

It is known that the *gamma* ($a, 1$) density has a monotone nondecreasing hazard rate when $a \geq 1$, and nonincreasing hazard rate when $a \leq 1$. The *Weibull* density

$$f(x) = ax^{a-1} \exp(-x^a), \quad x \geq 0,$$

has a nondecreasing hazard for $a \geq 1$, and a nonincreasing hazard rate for $a \leq 1$. For the exponential density e^{-x} , $x \geq 0$, we have a constant hazard rate 1, so that in (5), $x_i = i$, and thus,

$$E(N_s) = E(N_r) = \sum_{i=0}^{\infty} (1 - F(i)) = \sum_{i=0}^{\infty} e^{-i} = \frac{e}{e-1}.$$

For a survey of properties of distributions with monotone hazard rates, see Barlow, Marshall and Proschan (1963).

If we continue example 1 of sections 3.3 and 3.4 we see that for the beta $(1, a + 1)$ density ($a > 0$) $h(x) = \frac{a+1}{1-x}$, $0 \leq x \leq 1$, which is increasing on $[0, 1)$. Thus,

$E(N_s) = E(N_r) \leq \frac{e}{e-1}$ for all the examples mentioned above. For the beta $(1, a + 1)$ density, it is easy to derive an exact expression for $E(N_r)$ because (5) gives

$$x_0 = 0; x_{n+1} = x_n \left(\frac{a}{a+1} \right) + \frac{1}{a+1},$$

from which we obtain with some work that $x_n = 1 - \left(\frac{a}{a+1} \right)^n$, $n \geq 0$. Thus, by theorem 4,

$$E(N_s) = E(N_r) = \sum_{i=0}^{\infty} (1 - F(x_i)) = \sum_{i=0}^{\infty} (1 - x_i)^{a+1} = \sum_{i=0}^{\infty} \left(\frac{a}{a+1} \right)^{i(a+1)} = \frac{1}{1 - \left(\frac{1}{a+1} \right)^{a+1}},$$

which varies from 1 ($a=0$) to $\frac{e}{e-1}$ as $a \rightarrow \infty$ without exceeding $\frac{e}{e-1}$.

Consider next the Pareto density $f(x) = a/x^{a+1}$, $x \geq 1$. Here $F(x) = 1 - x^{-a}$, $x \geq 1$, and $h(x) = a/x$, $x \geq 1$. Also, the parameter a is positive. The bound in theorem 5 for nonincreasing h is very loose as we will see. Clearly, $X - 1$ is a monotone density bounded by $B = a$, $E(X - 1) = (a - 1)^{-1}$, $a > 1$, $E(X) = \infty$, $a \leq 1$. But (5) becomes $x_{n+1} = x_n \left(1 + \frac{1}{a} \right)$, so that $x_n = \left(1 + \frac{1}{a} \right)^n$ if the search is started at $x_0 = 1$. Thus,

$$E(N_s) = E(N_r) = \sum_{i=0}^{\infty} \left(1 + \frac{1}{a} \right)^{-ia} = \left(1 - \left(1 + \frac{1}{a} \right)^{-a} \right)^{-1},$$

and this varies from $\frac{e}{e-1}$ ($a \rightarrow \infty$), to 2 ($a=1$) and ∞ (as $a \downarrow 0$) asymptotic to

$$1 / \left(a \log \left(\frac{1}{a} \right) \right).$$

For most bounded monotone densities we will recommend inversion/rejection with Newton-Raphson iterations over inversion/rejection with doubling. Additional speed-ups are possible by storing for example a table of the first K constants $(x_n, f(x_n), F(x_n))$ (see remark 1). What is gained in time by the perfect balance $E(N_s) = E(N_r)$ usually offsets the loss due the fact that per iteration in the interval search one F computation and one f computation are needed versus only one F computation for the algorithm with doubling.

3.6. Table Methods

Let f be a bounded monotone density on $[0, 1]$, and assume that f can be computed but not F . Choose an integer $n \geq 1$ and note that f is dominated by g where

$$g(x) = f\left(\frac{i-1}{n}\right), \quad \frac{i-1}{n} \leq x < \frac{i}{n}, \quad i \text{ integer}, \quad 0 \leq x \leq 1.$$

The area under g is

$$c = \int_0^1 g(x) dx = \frac{1}{n} \sum_{i=1}^n f\left(\frac{i-1}{n}\right) \leq \frac{B}{n} + \int_0^1 f(x) dx = \frac{B}{n} + 1. \quad (6)$$

where $B = f(0)$ is the bound for f . By choosing n proportional to B (for example, $n = B$ or $n = 2B$). We see that the area under g stays uniformly bounded for all bounded monotone densities on $[0, 1]$! In fact, we can make this area as close to 1 as desired by choosing n large enough. Since random variates with density g can be obtained in constant average time by Walker's method (see Walker (1977) or Kronmal and Peterson (1979)), the ordinary rejection method with rejection from g yields a uniformly fast algorithm. In its simplest form, the *table method* can be summarized as follows.

Step 0: (Set-up.) Compute $f\left(\frac{i}{n}\right)$, $0 \leq i \leq n$, and store two tables:

$$p_i = f\left(\frac{i-1}{n}\right) \text{ and } q_i = f\left(\frac{i}{n}\right) / f\left(\frac{i-1}{n}\right), \quad 1 \leq i \leq n. \text{ Let } c = \frac{1}{n} \sum_{i=1}^n p_i.$$

Compute a table of aliases for Walker's method.

Step 1: Generate an integer I in the range $1 \leq I \leq n$ where $P(I=i) = p_i/(cn)$ by Walker's method. This requires average time bounded by a number that is independent of the p_i 's.

Generate two independent uniform $[0, 1]$ random variates U, V . Set $X \leftarrow (I-1 + V)/n$. (X has density proportional to g .)

Step 2: (Squeeze step. Optional, but recommended.) If $U \leq q_I$, exit with X . (Acceptance/rejection step.) If $U p_I \leq f(X)$, exit with X . Otherwise, go to 1.

Two quantities are of interest here:

1. N_r : the number of times that step 1 is executed.
2. N_q : the number of evaluations of f .

N_q is small when the squeeze step is efficient, and N_r is small if the dominating function g is close to f . The average time taken by the algorithm is obviously $0(E(N_r))$. Since the average time is also bounded from below by a constant times $E(N_r)$, we note that the influence of $E(N_q)$ must be limited to reducing the constant.

Theorem 6: *Let f be a monotone density on $[0, 1]$ bounded by B . For the table method, we have*

$$E(N_r) = c \leq 1 + \frac{B}{n}$$

and

$$\frac{B-1}{n} \leq E(N_q) \leq \frac{B}{n}.$$

Proof: Let f be a density, let g and h be two nonnegative functions, both integrable, let $0 \leq h \leq f \leq g$, and let the following rejection method be used to generate random variates with density f :

Step 1: Generate a random variate X with density proportional to g .
 Generate an independent uniform $[0, 1]$ random variate U .
 Set $T \leftarrow Ug(X)$.

Step 2: (Squeeze step.) If $T \leq h(X)$, exit with X .
 (Acceptance/rejection step.) If $T \leq f(X)$, exit with X . Otherwise, go to 1.

Then the expected number of executions of step 1 is $\int g$, and the expected number of evaluations of f is $\int (g - h)$, where all the integrals are with respect to dx . Thus, the first statement of theorem 6 follows from (6). The second statement follows from the observation that

$$E(N_q) = \sum_{i=1}^n \left(f\left(\frac{i-1}{n}\right) - f\left(\frac{i}{n}\right) \right) \frac{1}{n} = (f(0) - f(1)) \frac{1}{n} \in \left[\frac{B-1}{n}, \frac{B}{n} \right].$$

This concludes the proof of theorem 6.

When the table method is implemented, we should choose n proportional to B ; $n = B$ to $n = 20B$ seems to be the most useful range. Above $n = 20B$ the storage requirements become prohibitive. Since $E(N_q) \leq B/n$, we can eliminate the evaluations of f almost entirely, if we wish. By making n large enough, we can reduce the rejection rate at will: the optimal rejection rate of 0% can be approached. In other words, we are buying time with storage. A fair comparison between variable-storage methods (such as the table method) and fixed-storage methods (such as all the other methods discussed until now) is hardly possible: storage is usually cheap but rather inflexible, while time is expensive but available without limit. If for any physical reason one has to keep n smaller than K (say), then in view of $E(N_q) \geq (B-1)/n$, the average time becomes linear in B just as for the ordinary rejection method but with perhaps a smaller slope.

For the alias method we refer to Walker (1977). The table of n aliases needed in Walker's method can be found in time $O(n)$ (Kronmal and Peterson, 1979 a, b). Thus, step 0, the set-up step, requires time proportional to n . In practice, this means that the table method should be avoided in situations in which the number of random variates with the same density is smaller than n . If the density changes every k random variates for some fixed k , then the average time needed in the algorithm per random variate is bounded by

$$c_1 \left(1 + \frac{B}{n} \right) + c_2 \left(\frac{n}{k} \right)$$

where $c_1, c_2 > 0$ are constants. This is minimized by setting $n = \sqrt{c_1 Bk/c_2}$, and the upper bound becomes

$$c_1 + \sqrt{c_1 c_2 B/k}$$

which increases as \sqrt{B} . There is no hope of obtaining a better rate in B , and in this sense, the table method is doomed to be frustrating in many experiments.

Remark 6: [Modifications.]

The table method can be modified when additional information is known about f . Most of these modifications are detailed in Devroye (1982). The main conclusions are summarized here.

When f is *concave on its support*, the introduction of the squeeze step “If $U \leq 1 - V + Vq_T$, exit with X ” cuts $E(N_q)$ in half. Thus, we have

$$\frac{B-1}{2n} \leq E(N_q) \leq \frac{B}{2n}.$$

When f is *convex* the rectangular dominating curve g can be replaced by a piecewise linear curve g^* with breakpoints touching f . A quick analysis shows that $E(N_q)$ is cut in half. Also, by geometrical considerations,

$$E(N_q) = \frac{1}{2} \sum_{i=0}^{n-1} \frac{1}{n} f\left(\frac{i}{n}\right) + \frac{1}{2} \sum_{i=1}^n \frac{1}{n} f\left(\frac{i}{n}\right) \leq \frac{1}{2} \left(1 + \frac{B}{n}\right) + \frac{1}{2} = 1 + \frac{B}{2n}.$$

This is smaller than the upper bound of theorem 6.

We should emphasize that the table method given here does not require the computability of F . When F is also available, the algorithm can be modified to select the interval $\left[\frac{i-1}{n}, \frac{i}{n}\right)$ with the correct probability $F\left(\frac{i}{n}\right) - F\left(\frac{i-1}{n}\right)$. This results in a small gain in generation time for most distributions.

4. Experiments

We will illustrate all the algorithms with a simple yet flexible family of monotone densities on $[0, 1]$. Let

$$f(x) = \frac{\Gamma(a+b+1)}{\Gamma(a+1)\Gamma(b+1)} (1-x^{1/b})^a, \quad 0 \leq x \leq 1,$$

where $a, b > 0$ are parameters. The density f is *bounded* by

$$B = \frac{\Gamma(a+b+1)}{\Gamma(a+1)\Gamma(b+1)},$$

and B tends to ∞ when $a+b \rightarrow \infty$ and a, b stay bounded away from 0. Furthermore, f is *convex* if and only if $a \geq 1, b \geq 1$. It is *concave* if and only if $a \leq 1, b \leq 1$. The inverse of f is known explicitly:

$$f^{-1}(u) = \left(1 - \left(\frac{u}{B}\right)^{1/a}\right)^b, \quad 0 \leq u \leq B.$$

An explicit expression for F is only known in special cases, e. g. when a is integer. The family has many important limiting densities: the normal density ($b = \frac{1}{2}, a \rightarrow \infty$), the exponential density ($b = 1, a \rightarrow \infty$), the uniform density (a fixed, $b \downarrow 0$; or b fixed, $a \downarrow 0$) and the exponential power distributions (example 1 of section 3.3: b fixed, $a \rightarrow \infty$). Thus, it is flexible enough to be useful in a meaningful comparative experiment.

It is easy to check that X^b and $\left(\frac{Y}{Y+Z}\right)^b$ have density f , where X, Y and Z are independent beta($b, a+1$), gamma($b, 1$) and gamma($a+1, 1$) random variables respectively. In our experiments, we will only consider the special cases $a=1$ and $b=1$.

Theorem 7: For $a=1$, we obtain the density

$$f(x) = (b+1)(1-x^{1/b}), 0 \leq x \leq 1,$$

and the distribution function

$$F(x) = (b+1)x - b x^{1+1/b}, 0 \leq x \leq 1.$$

Random variates with this density can be generated as $UV^{b/(b+1)}$ where U, V are independent uniform $[0, 1]$ random variates. As $b \rightarrow \infty, f(x) \rightarrow \log\left(\frac{1}{x}\right), 0 \leq x \leq 1$.

Proof: The first statement follows from Khintchine's theorem (see e.g. Feller (1971), pp.158) for unimodal random variables. The last statement follows from the inequalities

$$\frac{b+1}{b} \log \frac{1}{x} \left(1 - \frac{1}{2b} \log \frac{1}{x}\right) \leq f(x) \leq \frac{b+1}{b} \log \frac{1}{x}, 0 \leq x \leq 1.$$

Theorem 8: For $b=1$, we obtain the density

$$f(x) = (a+1)(1-x)^a, 0 \leq x \leq 1.$$

The distribution function is $F(x) = 1 - (1-x)^{a+1}, 0 \leq x \leq 1$. Random variates with this density can be obtained in the following ways:

1. Generate a uniform $[0, 1]$ random variate U , and exit with $1 - U^{1/(a+1)}$.
2. Generate independent exponential and gamma ($a+1, 1$) random variates E and X , and exit with $\frac{E}{E+X}$.
3. Rejection form an exponential density:
 - Step 1: Generate two independent exponential random variates E_1, E_2 . Set $X \leftarrow E_1/a$. If $X > 1$, go to 1.
 - Step 2: If $E_2(1-X) - aX^2 \geq 0$, exit with X .
 - Step 3: If $aX + E_2 + a \log(1-X) \geq 0$, exit with X . Otherwise, go to 1.

4. *Rejection from a uniform density:*

Step 1: Generate two independent uniform $[0, 1]$ random variates U, X .

Step 2: If $U \leq (1 - X)^a$, exit with X . Otherwise, go to 1.

The average number of executions of step 1 in method 3 is $\frac{a+1}{a}$, and the average number of executions of step 1 in method 4 is $a + 1$. These average numbers are equal when $a=1$. The method that combines method 3 for $a \geq 1$ with method 4 for $a < 1$ has uniformly bounded average time; step 1 is guaranteed to be executed at most 2 times on the average.

Proof: No explanation is required, except for method 3, which is based on the inequalities

$$\exp(-ax/(1-x)) \leq (1-x)^a \leq \exp(-ax), \quad 0 \leq x \leq 1.$$

The different methods were coded in FORTRAN and compared on McGill University's AMDAHL V7 computer. For uniform $[0, 1]$ and exponential random variates, we used subprograms UNI and REXP of the "Super-duper" random number generator package. Only the density f given in the previous section was considered. The parameters a and b were varied as follows:

Experiment 1: $a = 1, b = 2^i/10, i = 2, \dots, 11$.

Experiment 2: $b = 1, a = 2^i/10, i = 2, \dots, 11$.

In all these cases, f is either convex or concave. Also, f has support $[0, 1]$ and is bounded by $B=(b+1)$ in Experiment 1, and by $B=(a+1)$ in Experiment 2. Necessary constants and tables are computed in separate set-up subprograms, and the density f or distribution function F is always passed as a parameter. The parameter passing slows the algorithms down, so that all the timings given here are pessimistic. No attempt was made to subtract the time due to overhead costs in subprogram calls. For each algorithm we will give:

1. The *average time* per random variate in microseconds. The average is obtained by repeating the experiment n times where n varied between 200 (for slow methods) and 5000 (fast methods). The results still have some residual variation, but we are not interested here in obtaining accurate times, but rather in detecting trends and making global comparisons.
2. The *set-up time* needed to compute all constants and tables.
3. The *size* of the compiled program (in bytes).
4. The *size* of a variable size *table* (in entries, or words).

For the FORTRAN programs we refer to Devroye (1982): these are available from the author upon request. The algorithms are given symbolic names in the tables:

D1: *direct method 1*: method 1 of theorem 8 for case $b = 1$.

D2: *direct method 2*: method 2 of theorem 8, with gamma variate generation by Marsaglia's squeeze method (Marsaglia, 1977). Case $b = 1$ only.

- D3: *direct method 3*: the combination of rejection method 3 and rejection method 4 of theorem 8. Case $b=1$ only.
- D4: *direct method 4*: the transformation-of-uniforms method suggested in theorem 7. Case $a=1$ only.
- D5: *direct method 5*: same as D4, but instead of $UV^{b/(b+1)}$, we use $U \exp(-Eb/(b+1))$ where U is uniform $[0, 1]$ and E is exponential. Case $a=1$ only.
- I: *inversion method* using Newton-Raphson iterations. See section 3.1.
- R1: *ordinary rejection method* with rectangular dominating function.
- R2: *modified rejection method* as described in section 3.2.
- IR1: *inversion/rejection method with halving*, $r=t=\frac{1}{2}$, see section 3.3.
- IR2: *inversion/rejection method with doubling*, $t=1/B$, $r=2$, see section 3.4.
- IR3: *inversion/rejection method with Newton-Raphson iterations*. See section 3.5.
- IR4: *inversion/rejection method with halving*, safe value for r .
- IR5: *inversion/rejection method with doubling*, safe value for r .
- $T(n)$: *table method with table size $\text{int}(nB)$* . Only values $n=1, 5$ and 20 are considered here. See section 3.6.

The direct methods are all uniformly fast over the variable parameters. They should not be considered as competitors, but as standards against which other performances are gauged. Our experiment demonstrates that no method except possibly the inversion method is strictly dominated by some other method in all respects. The inversion method seems dominated by the inversion/rejection method with Newton-Raphson iterations because the costly refinement is done by rejection in the latter method.

Table 1. Average times in experiment 1 as the parameter a varies from $2^2/10$ to $2^{11}/10$

D1	41.6	41.8	41.6	41.8	41.7	41.7	41.8	41.5	42.0	41.7
D2	44.7	43.4	44.3	43.5	43.2	41.3	41.2	43.2	40.7	41.3
D3	57.7	70.2	43.1	38.4	33.6	32.0	30.5	30.4	29.9	29.7
I	375.5	392.3	405.7	415.1	436.5	466.0	465.4	521.3	492.3	503.3
R1	70.7	83.5	128.9	187.7	341.9	596.4	1121	1836	4873	9310
R2	74.1	81.8	87.1	93.3	102.2	114.4	105.2	115.6	114.3	114.1
IR1	179.1	209.2	234.5	255.9	302.8	332.5	347.5	379.5	415.9	449.3
IR2	210.1	208.7	243.5	257.0	254.1	232.4	251.1	264.0	246.9	242.0
IR3	157.2	167.2	178.2	186.8	186.5	203.5	200.9	193.8	197.7	190.8
IR4	168.2	197.4	236.9	242.6	261.2	282.5	300.0	307.9	360.9	390.9
IR5	202.9	274.2	242.4	254.7	259.5	272.4	275.6	318.7	272.2	308.2
$T(1)$	73.8	73.8	100.2	86.8	87.8	86.3	89.4	87.3	88.7	88.5
$T(5)$	48.2	49.4	48.2	49.3	47.5	47.9	48.2	48.0	48.1	48.4
$T(20)$	40.8	40.9	40.8	41.9	41.0	40.9	41.7	41.5	—	—

Table 2. Average times in experiment 2 as the parameter b varies from $2^2/10$ to $2^{11}/10$

<i>D4</i>	45.3	45.3	45.3	45.4	45.3	45.3	45.3	45.3	45.4	45.3
<i>D5</i>	34.2	34.4	34.1	35.1	35.3	34.4	34.4	34.5	34.7	34.2
<i>I</i>	292.1	347.1	378.3	428.5	455.5	461.7	484.3	491.1	491.5	496.7
<i>R1</i>	73.1	85.6	138.1	188.3	376.6	634.6	1075	2028	4281	10100
<i>R2</i>	75.8	82.1	91.2	95.2	102.1	106.9	103.0	113.1	116.3	118.4
<i>IR1</i>	205.4	219.4	231.4	238.7	247.9	238.7	253.2	262.2	264.4	259.5
<i>IR2</i>	157.1	180.8	206.2	223.6	250.7	293.9	315.6	342.7	378.2	398.9
<i>IR3</i>	127.3	140.7	162.0	198.5	228.6	249.5	261.3	293.5	299.9	299.2
<i>IR4</i>	182.6	206.4	221.3	225.1	220.2	231.5	235.5	225.8	248.8	237.2
<i>IR5</i>	161.8	224.1	210.8	238.4	262.0	285.8	316.4	330.8	348.7	372.3
<i>T(1)</i>	73.7	73.7	100.1	86.0	80.2	74.2	69.4	66.5	69.1	64.6
<i>T(5)</i>	48.3	49.7	47.9	48.8	46.3	46.1	45.8	46.4	45.1	44.5
<i>T(20)</i>	40.9	40.9	40.7	41.5	40.6	40.8	40.2	40.6	—	—

Table 3. Properties of the various algorithms

	Average set-up time experiment 1	Average set-up time experiment 2	Program size (bytes)	Set-up program size (bytes)	Table size (words)
<i>D1</i>	0	—	402	0	0
<i>D2</i>	0	—	864	0	0
<i>D3</i>	0	—	650	0	0
<i>D4</i>	—	0	422	0	0
<i>D5</i>	—	0	416	0	0
<i>I</i>	0	0	642	0	0
<i>R1</i>	51	26	470	376	0
<i>R2</i>	71	51	794	498	0
<i>IR1</i>	0	0	632	0	0
<i>IR2</i>	0	0	686	0	0
<i>IR3</i>	0	0	718	0	0
<i>IR4</i>	70	50	654	474	0
<i>IR5</i>	70	50	708	508	0
<i>T(1)</i>	275.8 + 50.5 <i>B</i>	in both experiments	850	2142	int (<i>B</i>)
<i>T(5)</i>	244.6 + 257.8 <i>B</i>	in both experiments	850	2142	int (5 <i>B</i>)
<i>T(20)</i>	249.0 + 1142.3 <i>B</i>	in both experiments	850	2142	int (20 <i>B</i>)

Tables 1 and 2 show the average times per random variate for Experiments 1 and 2 respectively. Table 3 shows the other factors: the average set-up time, the size of the compiled program, and the table size. All the programs take between 402 and 864 bytes (with an extra 376 to 508 bytes for set-up programs with the exception of the set-up program for the table method, which requires 2142 bytes). Thus, except for the table method, the program size is not an important enough issue for deciding between algorithms. The true differentiation must be made on the basis of such factors as speed and flexibility.

Direct methods

Of the direct methods, *D3* is faster than *D1* and *D2*: *D1* uses costly operations all the time, and *D2* is indirect because we transform gamma random variates. Since the

original density f is simpler than the gamma density, such an indirect route can only lead to slowdowns.

$D5$ is faster than $D4$ because exponential random variates are generated faster than a logarithm is computed. This order may be reversed elsewhere.

Inversion method

In both experiments, the average time of the inversion method increases with B . No theoretical analysis of the average time was given in this paper because we wanted to avoid the messy issue of stopping times (should we stop when the absolute error is small, or when the relative error is small; and how are the errors obtained?). Such an analysis would be a waste of time because the method is obviously slower than all the other methods. In addition, it is the only non-exact method among all the methods considered here.

Rejection methods

The average times for $R1$, $R2$ are linear respectively logarithmic in B as was predicted by our analysis. The simple modification gives a dramatic improvement in performance. The improvement was so extraordinary that for the range of parameter values considered here, the modified rejection method was the fastest fixed storage method (the table method being the only variable storage method). It is recommended whenever f is monotone, bounded, and f has compact support. For unbounded monotone densities with unbounded support, we recommend a combination of several methods by partitioning the interval $[0, \infty)$ into $[0, 1]$ and $[1, \infty)$. For convex f (columns 3 through 10 in Tables 1, 2), we coded the improvement suggested in section 3.2, and obtained average times that varied from 80.5 (3rd column) to 102.5 (10th column). This is a 10% to 20% gain in average time.

Inversion/rejection methods

In all cases, we used the original algorithms of sections 3.3–3.5. None of the modifications suggested for convex or concave densities were implemented. From Examples 1 and 2 of sections 3.3–3.5 we recall that the average time in Experiment 1 increases as $\log(B)$ for $IR1$, and remains uniformly bounded for $IR2$ and $IR3$. This trend is observed in Table 1. In fact, $IR3$ is faster than $IR2$ which in turn is faster than $IR1$. $IR2$ and $IR3$ have similar left-to-right interval search components. Because of the perfect balance achieved by $IR3$, we expect $IR3$ to almost always perform better than $IR2$. The comparison with $IR1$ is not so straightforward, and it is for this reason that we choose to include Experiment 2 in this paper. For the density in Experiment 2 we have by elementary Taylor series inequalities

$$\frac{b+1}{b} \log\left(\frac{1}{x}\right) \geq f(x) = (b+1)(1-x^{1/b}) \geq \frac{b+1}{b} \log\left(\frac{1}{x}\right) \left(1 - \frac{1}{2b} \log\left(\frac{1}{x}\right)\right),$$

$$0 < x \leq 1.$$

Repeated use of $\int_0^1 \log^c \left(\frac{1}{x} \right) dx = \Gamma(c+1)$, $c > 0$, gives

$$\begin{aligned} 2 \frac{b+1}{b} \left(1 - \frac{3}{2b} \right) &= \int_0^1 \frac{b+1}{b} \left(\log^2 \left(\frac{1}{x} \right) - \frac{1}{2b} \log^3 \left(\frac{1}{x} \right) \right) dx \leq H(f) \\ &= \int_0^1 \left(\log \left(\frac{1}{x} \right) \right) f(x) dx \leq \min \left(b+1, 2 \frac{b+1}{b} \right) \leq 3. \end{aligned}$$

Obviously, $\sup_b H(f) \leq 3$, and $\lim_{b \rightarrow \infty} H(f) = 2$. By theorem 1 *IR1* has a small uniformly bounded average time. We will now show that the average time for *IR2* increases logarithmically in b as $b \rightarrow \infty$. By theorem 3 and our choice $\frac{1}{t} = B = b+1$, we need only show that $H^*(f) \sim \log(b)$ as $b \rightarrow \infty$. First, by an inequality derived in section 3.4,

$$\begin{aligned} H^*(f) &\leq \log(1 + E(BX)) \leq \log \left(1 + (b+1) \int_0^1 \frac{b+1}{b} x \log \left(\frac{1}{x} \right) dx \right) \\ &= \log \left(1 + \frac{(b+1)^2}{4b} \right) = (1 + o(1)) \log(b). \end{aligned}$$

A lower bound for $H^*(f)$ can be derived as follows: choose $\varepsilon > 0$ arbitrarily small, and let $c = \varepsilon + (1 - \varepsilon)/B$, $B = b+1$. Then

$$\begin{aligned} H^*(f) &= \int_0^1 \log_+(Bx) f(x) dx \geq \int_c^1 \log(Bc) f(x) dx \\ &= \log(Bc) (F(1) - F(c)) = \log(Bc) (1 - Bc + (B-1)c^{B/(B-1)}) \\ &= \log(Bc) (1 - c + (B-1)(c^{1+1/(B-1)} - c)) \\ &\geq \log(Bc) (1 - c + (B-1)c/(B-1)) \\ &= \log(Bc) \\ &= \log(B) + 0(1) \text{ as } B \rightarrow \infty. \end{aligned}$$

Thus, $H^*(f) \sim \log(b)$ as $b \rightarrow \infty$. Table 2 shows that *IR1* is indeed faster than *IR2* in Experiment 2. Once again, *IR3* is faster than *IR2*. *IR3* has increasing average times, but it is only for $b \geq 2^9/10$ that *IR1* becomes faster than *IR3*.

The choice $r=2$ in the halving and doubling methods can be replaced by a good guess of the optimal r . Based on the derivations of sections 3.4, 3.5 the following choices are suggested for r :

$$r = g(A) = \frac{A}{\log(A)} \left(\frac{1 + 2 \log(A)}{2 + \log(A)} \right), \quad A = 1 + \log(B) \quad \text{for } IR1,$$

$$r = \max(2, g(A)), \quad A = \log(1 + B) \quad \text{for } IR2.$$

These choices are by no means optimal. They only guarantee that the average times do not increase faster than $\log(B)/\log \log(B)$. They are applicable for all bounded

monotone densities because we can take $B=f(0)$. An improvement is expected for large values of B for those methods for which $H(f)$ or $H^*(f) \rightarrow \infty$ as $B \rightarrow \infty$ (IR1 in Experiment 1, IR2 in Experiment 2). A worsening is expected for the other cases, because a uniformly bounded average time is replaced by an unbounded average time increasing as $\log(B)/\log \log(B)$. These observations are corroborated by the timings for IR4 and IR5 in Tables 1 and 2.

Table methods

All the algorithms $T(n)$ have uniformly bounded average times. Very little improvement is possible beyond $n=20$. It is pleasing to see that we can approach the average times of the direct methods albeit by paying rather heavily in terms of storage (see Table 3). If we fix the table size, then the average time becomes linear in B . None of the improvements suggested in section 3.6 were implemented.

The rejection and table methods can only be used for bounded monotone densities with compact support, but they do not require the availability of F . The ordinary rejection method can in fact be considered as a table method with table size 1. It goes without saying that table methods should only be used when the density changes infrequently because of the prohibitive set-up times involved.

We conclude by noting that in terms of flexibility, the inversion/rejection methods have no competition: a suitable combination of two of them can be used for *all* monotone densities. The halving method takes care of the peak at 0 while the doubling method or the Newton-Raphson iterations could be used to handle infinite tails. The only restriction is that both f and F must be computable.

Acknowledgement

The author acknowledges the helpful suggestions of one referee.

References

- Ahrens, J. H., Kohrt, K. D.: Computer methods for efficient sampling from largely arbitrary statistical distributions. *Computing* 26, 19–31 (1981).
- Barlow, R. E., Marshall, A. W., Proschan, F.: Properties of probability distributions with monotone hazard rate. *Annals of Mathematical Statistics* 34, 375–389 (1963).
- Chen, H. C., Asau, Y.: On generating random variates from an empirical distribution. *AIIE Transactions* 6, 163–166 (1974).
- Devroye, L.: Random variate generation for unimodal and monotone densities. Technical Report, School of Computer Science, McGill University, Montreal, Canada H3A 2K6, 1982.
- Feller, W.: *An Introduction to Probability Theory and Its Applications*, Vol. 1. New York: J. Wiley 1961.
- Feller, W.: *An Introduction to Probability Theory and Its Applications*, Vol. 2, 2nd ed. New York: J. Wiley 1971.
- Hardy, G. H., Littlewood, J. E., Polya, G.: *Inequalities*, 2nd ed. London: Cambridge University Press 1952.
- Kohrt, K. D.: Efficient sampling from non-uniform statistical distributions. Diploma Thesis, University of Kiel, Federal Republic of Germany, 1980.
- Kronmal, R. A., Peterson, A. V.: On the alias method for generating random variables from a discrete distribution. *The American Statistician* 33, 214–218 (1979a).

- Kronmal, R. A., Peterson, A. V.: Programs for generating discrete random integers using Walker's alias method. Manuscript, Department of Biostatistics, University of Washington, Seattle, Washington, 1979b.
- Marsaglia, G.: The squeeze method for generating gamma variates. *Computers and Mathematics with Applications* 3, 321–325 (1977).
- Schmeiser, B. W.: Random variate generation: a survey. Proceedings of the 1980 Winter Simulation Conference, Orlando, Florida, 1980.
- Steffensen, J. F.: On certain inequalities between mean values and their application to actuarial problems. *Skand. Aktuarietidskrift* 1, 82–97 (1918).
- Steffensen, J. F.: On a generalization of certain inequalities of Tchebychef and Jensen. *Skand. Aktuarietidskrift* 8, 137–147 (1925).
- Walker, A. J.: An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software* 3, 253–256 (1977)

Prof. Dr. L. Devroye
School of Computer Science
McGill University
805 Sherbrooke Street West
Montreal, P. Q.
Canada, H3A 2K6