Chapter Two
GENERAL PRINCIPLES IN RANDOM VARIATE GENERATION

1. INTRODUCTION.

In this chapter we introduce the reader to the fundamental principles in non-uniform random variate generation. This chapter is a must for the serious reader. On its own it can be used as part of a course in simulation.

These basic principles apply often, but not always, to both continuous and discrete random variables. For a structured development it is perhaps best to develop the material according to the guiding principle rather than according to the type of random variable involved. The reader is also cautioned that we do not make any recommendations at this point about generators for various distributions. All the examples found in this chapter are of a didactical nature, and the most important families of distributions will be studied in chapters IX,X,XI in more detail.

2. THE INVERSION METHOD.

2.1. The inversion principle.

The inversion method is based upon the following property:

Theorem 2.1.

Let F be a continuous distribution function on R with inverse F^{-1} defined by

$$F^{-1}(u) = \inf \{x : F(x) = u, 0 < u < 1\}.$$

If U is a uniform [0,1] random variable, then $F^{-1}(U)$ has distribution function F. Also, if X has distribution function F, then $F(X)$ is uniformly distributed on [0,1].

Proof of Theorem 2.1.

The first statement follows after noting that for all $x \in R$,

$$P(F^{-1}(U) \leq x) = P(\inf \{y : F(y) = U\} \leq x)$$
$$= P(U \leq F(x)) = F(x).$$

The second statement follows from the fact that for all $0 < u < 1$,

$$P(F(X) \leq u) = P(X \leq F^{-1}(u))$$
$$= F(F^{-1}(u)) = u \quad \blacksquare.$$

Theorem 2.1 can be used to generate random variates with an arbitrary continuous distribution function F provided that F^{-1} is explicitly known. The faster the inverse can be computed, the faster we can compute X from a given uniform [0,1] random variate U. Formally, we have

The inversion method

Generate a uniform [0,1] random variate U.
RETURN $X \leftarrow F^{-1}(U)$

In the next table, we give a few important examples. Often, the formulas for

$F^{-1}(U)$ can be simplified, by noting for example that $1-U$ is distributed as U.

Density $f(x)$	$F(x)$	$X=F^{-1}(U)$	Simplified form
Exponential(λ) $\lambda e^{-\lambda x}$, $x \geq 0$	$1-e^{-\lambda x}$	$-\frac{1}{\lambda}\log(1-U)$	$-\frac{1}{\lambda}\log(U)$
Cauchy(σ) $\frac{\sigma}{\pi(x^2+\sigma^2)}$	$\frac{1}{2}+\frac{1}{\pi}\arctan(\frac{x}{\sigma})$	$\sigma\tan(\pi(U-\frac{1}{2}))$	$\sigma\tan(\pi U)$
Rayleigh(σ) $\frac{x}{\sigma}e^{-\frac{x^2}{2\sigma^2}}$, $x \geq 0$	$1-e^{-\frac{x^2}{2\sigma^2}}$	$\sigma\sqrt{-\log(1-U)}$	$\sigma\sqrt{-\log(U)}$
Triangular on$(0,a)$ $\frac{2}{a}(1-\frac{x}{a})$, $0 \leq x \leq a$	$\frac{2}{a}(x-\frac{x^2}{2a})$	$a(1-\sqrt{1-U})$	$a(1-\sqrt{U})$
Tail of Rayleigh $xe^{\frac{a^2-x^2}{2}}$, $x \geq a > 0$	$1-e^{\frac{a^2-x^2}{2}}$	$\sqrt{a^2-2\log(1-U)}$	$\sqrt{a^2-2\log U}$
Pareto(a,b) $\frac{ab^a}{x^{a+1}}$, $x \geq b > 0$	$1-(\frac{b}{x})^a$	$\frac{b}{(1-U)^{1/a}}$	$\frac{b}{U^{1/a}}$

There are many areas in random variate generation where the inversion method is of particular importance. We cite four examples:

Example 2.1. Generating correlated random variates.

When two random variates X and Y are needed with distribution functions F and G respectively, then these can be obtained as $F^{-1}(U)$ and $G^{-1}(V)$ where U and V are uniform [0,1] random variates. If U and V are dependent, then so are $F^{-1}(U)$ and $G^{-1}(V)$. Maximal correlation is achieved by using $V=U$, and maximal negative correlation is obtained by setting $V=-U$. While other methods may be available for generating X and/or Y individually, few methods allow the flexibility of controlling the correlation as described here. In variance reduction, negatively correlated random variates are very useful (see e.g. Hammersley and Handscomb, 1964, or Bratley, Fox and Schrage, 1984). ■

Example 2.2. Generating maxima.

To generate $X = \max(X_1, \ldots, X_n)$, where the X_i's are iid random variates with distribution function F, we could:

(i) Generate $X_1, \ldots, X_n$, and take the maximum.

(ii) Generate a uniform [0,1] random variate U and find the solution X of $F^n(X) = U$.

(iii) Generate V, a random variate distributed as the maximum of n iid uniform [0,1] random variates, and find the solution X of $F(X) = V$.

Thus, the elegant solutions (ii) and (iii) involve inversion. ■

Example 2.3. Generating all order statistics.

A sample $X_{(1)}, \ldots, X_{(n)}$ of order statistics of a sequence $X_1, \ldots, X_n$ of iid random variables with distribution function F can be obtained as $F^{-1}(U_{(1)}), \ldots, F^{-1}(U_{(n)})$, where the $U_{(i)}$'s are the order statistics of a uniform sample. As we will see further on, this is often more efficient than generating the X_i sample and sorting it. ■

Example 2.4. A general purpose generator.

The inversion method is the only truly universal method: if all we can do is compute $F(x)$ for all x, and we have enough (i.e., infinite) time on our hands, then we can generate random variates with distribution function F. All the other methods described in this book require additional information in one form or another. ■

2.2. Inversion by numerical solution of F(X)=U.

The inversion method is exact when an explicit form of F^{-1} is known. In other cases, we must solve the equation $F(X) = U$ numerically, and this requires an infinite amount of time when F is continuous. Any stopping rule that we use with the numerical method leads necessarily to an inexact algorithm. In this section we will briefly describe a few numerical inversion algorithms and stopping rules. Despite the fact that the algorithms are inexact, there are situations in which we are virtually forced to use numerical inversion, and it is important to compare different inversion algorithms from various points of view.

In what follows, X is the (unknown, but exact) solution of $F(X) = U$, and $X*$ is the value returned by the numerical inversion algorithm. A stopping rule which insists that $|X*-X| < \delta$ for some small $\delta>0$ is not realistic because for large values of X, this would probably imply that the number of significant digits is greater than the built-in limit dictated by the wordsize of the computer. A second choice for our stopping rule would by $|F(X*)-F(X)| < \epsilon$, where $\epsilon>0$ is a small number. Since all F values are in the range [0,1], we do not face the above-mentioned problem any more, were it not for the fact that small variations in X can lead to large variations in $F(X)$-values. Thus, it is possible that even the smallest realizable increment in X yields a change in $F(X)$ that exceeds the given constant ϵ. A third possibility for our stopping rule would be $|X*-X| < \delta |X|$ where the value of δ is determined by the wordsize of the computer. While this addresses the problem of relative accuracy correctly, it will lead to more accuracy than is orinarily required for values of X near 0. Thus, no stopping rule seems universally recommendable. If we know that X takes values in [-1,1], then the rule $|X*-X| < \delta$ seems both practical and amenable to theoretical analysis. Let us first see what we could do when the support of F falls outside [-1,1].

Let $h : R \to (-1,1)$ be a strictly monotone continuous transformation. Assume now that we obtain $X*$ by the following method:

> Let $Y*$ be the numerical solution of $F(h^{-1}(y)) = U$, where U is a uniform [0,1] random variable and $Y*$ is such that it is within δ of the exact solution Y of the given equation.
> $X* \leftarrow h^{-1}(Y*)$

Here we used the fact that Y has distribution function $F(h^{-1}(y))$, $|y| \leq 1$. Let us now look at what happens to the accuracy of the solution. A variation of dy on the value of y leads to variation of $h^{-1\prime}(y)\,dx = h^{-1\prime}(h(x))\,dx$ on the corresponding value of x. The expected variation thus is about equal to $V\delta$ where

$$V = E(h^{-1\prime}(h(X))) = E\left(\frac{1}{h'(X)}\right).$$

Unfortunately, the best transformation h, i.e. the one that minimizes V, depends upon the distribution of X. We can give the reader some insight in how to choose h by an example. Consider for example the class if transformations

$$h(x) = \frac{x-m}{s+|x-m|},$$

where $s>0$ and $m \in R$ are constants. Thus, we have $h^{-1}(y) = m+sy/(1-|y|)$, and

$$V = E(\frac{1}{s}(s+|X-m|)^2) = s+2E(|X-m|)+\frac{1}{s}E((X-m)^2).$$

For symmetric random variables X, this expression is minimized by setting $m=0$ and $s=\sqrt{Var(X)}$. For asymmetric X, the minimization problem is very difficult. The next best thing we could do is minimize a good upper bound for V, such as the one provided by applying the Cauchy-Schwarz inequality,

$$V \leq s+2\sqrt{E(X-m)^2)}+\frac{1}{s}E((X-m)^2).$$

This upper bound is minimal when

$$m = E(X), \; s = \sqrt{Var(X)}.$$

The upper bound for V then becomes $4\sqrt{Var(X)}$. This approach requires either exact values or good approximations for m and s. We refer to Exercise 1 for a detailed comparison of the average accuracy of this method with that of the direct solution of $F(X) = U$ given that the same stopping rule is used.

We will discuss three popular numerical inversion algorithms for $F(X)=U$:

The bisection method

Find an initial interval $[a,b]$ to which the solution belongs.
REPEAT
    $X \leftarrow (a+b)/2$
    IF $F(X) \leq U$
        THEN $a \leftarrow X$
        ELSE $b \leftarrow X$
UNTIL $b-a \leq 2\delta$
RETURN X

The secant method (regula falsi method)

Find an interval $[a,b]$ to which the solution belongs.
REPEAT
$X \leftarrow a+(b-a)\dfrac{U-F(a)}{F(b)-F(a)}$
IF $F(X) \leq U$
THEN $a \leftarrow X$
ELSE $b \leftarrow X$
UNTIL $b-a \leq \delta$
RETURN X

The Newton-Raphson method

Choose an initial guess X.
REPEAT
$X \leftarrow X-\dfrac{(F(X)-U)}{f(X)}$
UNTIL stopping rule is satisfied. (Note: f is the density corresponding to F.)
RETURN X.

In the first two methods, we need an initial interval $[a,b]$ known to contain the solution. If the user knows functions G and H such that $G(x) \geq F(x) \geq H(x)$ for all x, then we could start with $[a,b] = [G^{-1}(U), H^{-1}(U)]$. In particular, if the support of F is known, then we can set $[a,b]$ equal to it. Because it is important to have reasonably small intervals, any a priori information should be used to select $[a,b]$. For example, if F has variance σ^2 and is symmetric about 0, then by Cantelli's extension of Chebyshev's inequality,

$$F(x) \geq \frac{x^2}{x^2+\sigma^2} \quad (x>0).$$

This suggests that when $U > \frac{1}{2}$, we take

$$[a,b] = [0, \sigma\sqrt{\frac{U}{1-U}}].$$

When $U \leq \frac{1}{2}$, we argue by symmetry. Thus, information about moments and quantiles of F can be valuable for initial guesswork. For the Newton-Raphson method, we can often take an arbitrary point such as 0 as our initial guess.

The actual choice of an algorithm depends upon many factors such as

(i) Guaranteed convergence.
(ii) Speed of convergence.
(iii) A priori information.
(iv) Knowledge of the density f .

If f is not explicitly known, then the Newton-Raphson method should be avoided because the approximation of $f(x)$ by $\frac{1}{\delta}(F(x+\delta)-F(x))$ is rather inaccurate because of cancelation errors.

Only the bisection method is guaranteed to converge in all cases. If $F(X)=U$ has a unique solution, then the secant method converges too. By "convergence" we mean of course that the returned variable $X*$ would approach the exact solution X if we would let the number of iterations tend to ∞. The Newton-Raphson method converges when F is convex or concave. Often, the density f is unimodal with peak at m. Then, clearly, F is convex on $(-\infty,m]$, and concave on $[m,\infty)$, and the Newton-Raphson method started at m converges.

Let us consider the speed of convergence now. For the bisection method started at $[a,b] = [g_1(U),g_2(U)]$ (where g_1,g_2 are given functions), we need N iterations if and only if

$$2^{N-1} < g_2(U)-g_1(U) \leq 2^N .$$

The solution of this is

$$N = 1+\left\lfloor \log_+((g_2(U)-g_1(U))/\delta) \right\rfloor ,$$

where $\log_+$ is the positive part of the logarithm with base 2. From this expression, we retain that $E(N)$ can be infinite for some long-tailed distributions. If the solution is known to belong to $[-1,1]$, then we have deterministically,

$$N \leq 1+\log_+(\frac{1}{\delta}) .$$

And in all cases in which $E(N) < \infty$, we have as $\delta \downarrow 0$, $E(N) \sim \log(\frac{1}{\delta})$. Essentially, adding one bit of accuracy to the solution is equivalent to adding one iteration. As an example, let us take $\delta = 10^{-7}$, which corresponds to the standard choice for problems with solutions in $[-1,1]$ when a 32-bit computer is used. The value of N in that case is in the neighborhood of 24, and this is often inacceptable.

The secant and Newton-Raphson methods are both faster, albeit less robust, than the bisection method. For a good discussion of the convergence and rate of convergence of the given methods, we refer to Ostrowski (1973). Let us merely

state one of the results for $E(N)$, the quantity of interest to us, where N is the number of iterations needed to get to within δ of the solution (note that this is impossible to verify when an algorithm is running !). Also, let F be the distribution function corresponding to a unimodal density with absolutely bounded derivative f'. The Newton-Raphson method started at the mode converges, and for some number N_0 depending only upon F (but possibly ∞) we have

$$E(N) \leq N_0 + \text{loglog}(\frac{1}{\delta})$$

where all logarithms are base 2. For the secant method, a similar statement can be made but the base should be replaced by the golden ratio, $\frac{1}{2}(1+\sqrt{5})$. In both cases, the influence of δ on the average number of iterations is practically nil, and the asymptotic expression for $E(N)$ is smaller than in the bisection method (when $\delta \downarrow 0$). Obviously, the secant and Newton-Raphson methods are not universally faster than the bisection method. For ways of accelerating these methods, see for example Ostrowski (1973, Appendix I, Appendix G).

2.3. Explicit approximations.

When F^{-1} is not explicitly known, it can sometimes be well approximated by another explicitly known function $g(U)$. In iterative methods, the stopping rule usually takes care of the accuracy problem. Now, by resorting to a one-step procedure, we squarely put the burden of verifying the accuracy of the solution on the shoulders of the theoretician. Also, we should define once again what we mean by accuracy (see Devroye (1982) for a critical discussion of various definitions). Iterative methods can be notoriously slow, but this is a small price to pay for their conciseness, simplicity, flexibility and accuracy. The four main limitations of the direct approximation method are:

(i) The approximation is valid for a given F: to use it when F changes frequently during the simulation experiment would probably require extraordinary set-up times.

(ii) The function g must be stored. For example, g is often a ratio of two polynomials, in which case all the coefficients must be put in a long table.

(iii) The accuracy of the approximation is fixed. If a better accuracy is needed, the entire function g must be replaced. This happens for example when one switches to a computer with a larger wordsize. In other words, future computer upgrades will be expensive.

(iv) Certain functions cannot be approximated very well by standard approximation techniques, except possibly by inacceptably complicated functions. Also, approximations are difficult to develop for multiparameter families of functions.

How one actually goes about designing approximations g will not be explained here. For example, we could start from a very rough approximation of F^{-1}, and then explicitly compute the function that corresponds to one or two or a fixed number of Newton-Raphson iterations. This is not systematic enough in general. A spline method was developed in Kohrt (1980) and Ahrens and Kohrt (1981). In the general literature, one can find many examples of approximations by ratios of polynomials. For example, for the inverse of the normal distribution function, Odeh and Evans (1974) suggest

$$g(u) = \sqrt{-2\log(u)} + \frac{A(\sqrt{-2\log(u)})}{B(\sqrt{-2\log(u)})}, \frac{1}{2} \geq u \geq 10^{-20},$$

where $A(x) = \sum_{i=0}^{4} a_i x^i$, and $B(x) = \sum_{i=0}^{4} b_i x^i$, and the coefficients are as shown in the table below:

i	a_i	b_i
0	-0.322232431088	0.0993484626060
1	-1.0	0.588581570495
2	-0.342242088547	0.531103462366
3	-0.0204231210245	0.103537752850
4	-0.0000453642210148	0.0038560700634

For u in the range $[\frac{1}{2}, 1-10^{-20}]$, we take $-g(1-u)$, and for u in the two tiny left-over intervals near 0 and 1, the approximation should not be used. Rougher approximations can be found in Hastings (1955) and Bailey (1981). Bailey's approximation requires fewer constants and is very fast. The approximation of Beasley and Springer (1977) is also very fast, although not as accurate as the Odeh-Evans approximation given here. Similar methods exist for the inversion of beta and gamma distribution functions.

2.4. Exercises.

1. Most stopping rules for the numerical iterative solution of $F(X)=U$ are of the type $b-a \leq \delta$ where $[a,b]$ is an interval containing the solution X, and $\delta>0$ is a small number. These algorithms may never halt if for some u, there is an interval of solutions of $F(X)=u$ (this applies especially to the secant method). Let A be the set of all u for which we have for some $x<y$, $F(x)=F(y)=u$. Show that $P(U \in A)=0$, i.e. the probability of ending up in an infinite loop is zero. Thus, we can safely lift the restriction imposed throughout this section that $F(X)=u$ has one solution for all u.
2. Show that the secant method converges if $F(X)=U$ has one solution for the given value of U.
3. Show that if $F(0)=0$ and F is concave on $[0,\infty)$, then the Newton-Raphson method started at 0 converges.

4. **Student's t distribution with 3 degrees of freedom.**
Consider the density

$$f(x) = \frac{2}{\pi(1+x^2)^2},$$

and the corresponding distribution function

$$F(x) = \frac{1}{2}+\frac{1}{\pi}(\arctan x + \frac{x}{1+x^2}).$$

These functions define the t distribution with 3 degrees of freedom. Elsewhere we will see very efficient methods for generating random variates from this distribution. Nevertheless, because F^{-1} is not known explicitly (except perhaps as an infinite series), this distribution can be used to illustrate many points made in the text. Note first that the distribution is symmetric about 0. Prove first that

$$\frac{1}{2}+\frac{1}{\pi}\arctan x \le F(x) \le \frac{1}{2}+\frac{2}{\pi}\arctan x \quad (x \ge 0).$$

Thus, for $U \ge \frac{1}{2}$, the solution of $F(X)=U$ lies in the interval

$$[\tan(\frac{\pi}{2}(U-\frac{1}{2})),\tan(\pi(U-\frac{1}{2}))].$$

Using this interval as a starting interval, compare and time the bisection method, the secant method and the Newton-Raphson method (in the latter method, start at 0 and keep iterating until X does not change in value any further). Finally, assume that we have an efficient Cauchy random variate generator at our disposal. Recalling that a Cauchy random variable C is distributed as $\tan(\pi(U-\frac{1}{2}))$, show that we can generate X by solving the equation

$$\arctan X + \frac{X}{1+X^2} = \arctan C,$$

and by starting with initial interval

$$[\sqrt{\frac{\sqrt{1+C^2}-1}{\sqrt{1+C^2}+1}}, C]$$

when $C>0$ (use symmetry in the other case). Prove that this is a valid method.

5. Develop a general purpose random variate generator which is based upon inversion by the Newton-Raphson method, and assumes only that F and the corresponding density f can be computed at all points, and that f is unimodal. Verify that your method is convergent. Allow the user to specify a mode if this information is available.

6. Write general purpose generators for the bisection and secant methods in which the user specifies an initial interval $[g_1(U), g_2(U)]$.

7. Discuss how you would solve $F(X)=U$ for X by the bisection method if no initial interval is available. In a first stage, you could look for an interval $[a, b]$ which contains the solution X. In a second stage, you proceed by ordinary bisection until the interval's length drops below δ. Show that regardless of how you organize the original search (this could be by looking at adjacent intervals of equal length, or adjacent intervals with geometrically increasing lengths, or adjacent intervals growing as $2, 2^2, 2^{2^2}, \ldots$), the expected time taken by the entire algorithm is ∞ whenever $E(\log_+ |X|)=\infty$. Show that for extrapolatory search, it is not a bad strategy to double the interval sizes. Finally, exhibit a distribution for which the given expected search time is ∞. (Note that for such distributions, the expected number of bits needed to represent the integer portion is infinite.)

8. **An exponential class of distributions.** Consider the distribution function $F(x)=1-e^{-A_n(x)}$ where $A_n(x)=\sum_{i=1}^{n} a_i x^i$ for $x \geq 0$ and $A_n(x)=0$ for $x<0$. Assume that all coefficients a_i are nonnegative and that $a_1>0$. If U is a uniform [0,1] random variate, and E is an exponential random variate, then it is easy to see that the solution of $1-e^{-A_n(X)}=U$ is distributed as the solution of $A_n(X)=E$. The basic Newton-Raphson step for the solution of the second equation is

$$X \leftarrow X - \frac{A_n(X)-E}{A_n'(X)} .$$

Since $a_1>0$ and A_n is convex, any starting point $X \geq 0$ will yield a convergent sequence of values. We can thus start at $X=0$ or at $X=E/a_1$ (which is the first value obtained in the Newton-Raphson sequence started at 0). Compare this algorithm with the algorithm in which X is generated as

$$\min_{1 \leq i \leq n} \left(\frac{E_i}{a_i}\right)^{\frac{1}{i}}$$

where $E_1, \ldots, E_n$ are iid exponential random variates.

9. **Adaptive inversion.** Consider the situation in which we need to generate a sequence of n iid random variables with continuous distribution function F by the method of inversion. The generated couples $(X_1, U_1), \ldots$ are stored ($X_1=F^{-1}(U_1)$ and U_1 is uniform [0,1]). Define an algorithm based upon a dynamic hash table for the U_i's in which the table is used to find a good starting interval for inversion. Implement, and compare this adaptive method with memoryless algorithms (Yuen, 1981).

10. **Truncated distributions.** Let X be a random variable with distribution function F. Define the truncated random variable Y by its distribution

function

$$G(x) = \begin{cases} 0 & x < a \\ \dfrac{F(x)-F(a)}{F(b)-F(a)} & a \le x \le b \\ 1 & x > b \end{cases}.$$

Here $-\infty \le a < b \le \infty$. Show that Y can be generated as $F^{-1}(F(a)+U(F(b)-F(a)))$ where U is a uniform [0,1] random variate.

11. Find a monotonically decreasing density f on $[0,\infty)$ such that the Newton-Raphson procedure started at $X=0$ needs N steps to get within δ of the solution of $F(X)=U$ where N is a random variable with mean $E(N)=\infty$ for all $\delta>0$.

12. **The logistic distribution.** A random variable X is said to have the logistic distribution with parameters $a \in R$ and $b > 0$ when

$$F(x) = \frac{1}{1+e^{-\frac{x-a}{b}}}.$$

It is obvious that a is a translation parameter and that b is a scale parameter. The standardized logistic distribution has $a=0, b=1$. The density is

$$f(x) = \frac{e^{-x}}{(1+e^{-x})^2} = F(x)(1-F(x)).$$

The logistic density is symmetric about 0 and resembles in several respects the normal density. Show the following:

A. When U is uniformly distributed on [0,1], then $X=\log(\frac{U}{1-U})$ has the standard logistic distribution.

B. $\frac{U}{1-U}$ is distributed as the ratio of two iid exponential random variables.

C. We say that a random variable Z has the extremal value distribution with parameter a when $F(x)=e^{-ae^{-x}}$. If X is distributed as Z with parameter Y where Y is exponentially distributed, then X has the standard logistic distribution.

D. $E(X^2)=\frac{\pi^2}{3}$, and $E(X^4)=\frac{7\pi^4}{15}$.

E. If X_1, X_2 are independent extremal value distributed random variables with the same parameter a, then X_1-X_2 has a logistic distribution.

3. THE REJECTION METHOD.

3.1. Definition.

The rejection method is based upon the following fundamental property of densities:

Theorem 3.1.

Let X be a random vector with density f on R^d, and let U be an independent uniform [0,1] random variable. Then $(X, cUf(X))$ is uniformly distributed on $A = \{(x,u): x \in R^d, 0 \le u \le cf(x)\}$, where $c > 0$ is an arbitrary constant. Vice versa, if (X,U) is a random vector in R^{d+1} uniformly distributed on A, then X has density f on R^d.

Proof of Theorem 3.1.

For the first statement, take a Borel set $B \subseteq A$, and let B_x be the section of B at x, i.e. $B_x = \{u: (x,u) \in B\}$. By Tonelli's theorem,

$$P((X, cUf(X)) \in B) = \int\int_{B_x} \frac{1}{cf(x)}\, du\ f(x)\, dx = \frac{1}{c}\int_B du\ dx .$$

Since the area of A is c, we have shown the first part of the Theorem. The second part follows if we can show that for all Borel sets B of R^d, $P(X \in B) = \int_B f(x)\, dx$ (recall the definition of a density). But

$$P(X \in B) = P((X,U) \in B_1 = \{(x,u): x \in B, 0 \le u \le cf(x)\})$$

$$= \frac{\int\int_{B_1} du\ dx}{\int\int_A du\ dx} = \frac{1}{c}\int_B cf(x)\, dx = \int_B f(x)\, dx ,$$

which was to be shown. ■

Theorem 3.2.

Let $X_1, X_2, \ldots$ be a sequence of iid random vectors taking values in R^d, and let $A \subseteq R^d$ be a Borel set such that $P(X_1 \in A) = p > 0$. Let Y be the first X_i taking values in A. Then Y has a distribution that is determined by

$$P(Y \in B) = \frac{P(X_1 \in A \cap B)}{p}, \quad B \text{ Borel set of } R^d .$$

In particular, if X_1 is uniformly distributed in A_0 where $A_0 \supseteq A$, then Y is uniformly distributed in A.

Proof of Theorem 3.2.

For arbitrary Borel sets B, we observe that

$$\begin{aligned} P(Y \in B) &= \sum_{i=1}^{\infty} P(X_1 \notin A, \ldots, X_{i-1} \notin A, X_i \in B \cap A) \\ &= \sum_{i=1}^{\infty} (1-p)^{i-1} P(X_1 \in A \cap B) \\ &= \frac{1}{1-(1-p)} P(X_1 \in A \cap B), \end{aligned}$$

which was to be shown. If X_1 is uniformly distributed in A_0, then

$$P(Y \in B) = \frac{P(X_1 \in A \cap B)}{P(X_1 \in A)} = \frac{\int_{A_0 AB} dx}{\int_{A_0} dx} \cdot \frac{\int_{A_0} dx}{\int_{AA_0} dx} = \frac{\int_{AB} dx}{\int_{A} dx} .$$

This concludes the proof of Theorem 3.2. ■

The basic version of the rejection algorithm assumes the existence of a density g and the knowledge of a constant $c \geq 1$ such that

$$f(x) \leq c g(x) \quad \text{(all } x\text{)} .$$

Random variates with density f on R^d can be obtained as follows:

The rejection method

REPEAT

Generate two independent random variates X (with density g on R^d) and U (uniformly distributed on [0,1]).

Set $T \leftarrow c\frac{g(X)}{f(X)}$.

UNTIL $UT \leq 1$

RETURN X

By Theorem 3.1, $(X, cUg(X))$ (where X and U are as explained in the first line of the REPEAT loop) is uniformly distributed under the curve of cg in R^{d+1}. By Theorem 3.2, we conclude that the random variate $(X, cUg(X))$ generated by this algorithm (i.e. at time of exit) is uniformly distributed under the curve of f. By the second part of Theorem 3.1, we can then conclude that its d-dimensional projection X must have density f.

The three things we need before we can apply the rejection algorithm are (i) a dominating density g; (ii) a simple method for generating random variates with density g; and (iii) knowledge of c. Often, (i) and (iii) can be satisfied by a priori inspection of the analytical form of f. Basically, g must have heavier tails and sharper infinite peaks than f. In some situations, we can determine cg for entire classes of densities f. The dominating curves cg should always be picked with care: not only do we need a simple generator for g (requirement (ii)), but we must make sure that the computation of $\frac{g(X)}{f(X)}$ is simple. Finally, cg must be such that the algorithm is efficient.

Let N be the number of iterations in the algorithm, i.e. the number of pairs (X, U) required before the algorithm halts. We have

$$P(N=i) = (1-p)^{i-1}p \; ; P(N \geq i) = (1-p)^{i-1} \quad (i \geq 1),$$

where

$$p = P(f(X) \geq cUg(X)) = \int P(U \leq \frac{f(x)}{cg(x)})\, dx$$

$$= \int \frac{f(x)}{cg(x)}\, g(x)\, dx = \frac{1}{c}\int f(x)\, dx = \frac{1}{c}.$$

Thus, $E(N) = \frac{1}{p} = c$, $E(N^2) = \frac{2}{p^2} - \frac{1}{p}$ and $Var(N) = \frac{1-p}{p^2} = c^2 - c$. In other words, $E(N)$ is one over the probability of accepting X. From this we conclude that we should keep c as small as possible. Note that the distribution of N is geometric with parameter $p = \frac{1}{c}$. This is good, because the probabilities

$P(N=i)$ decrease monotonically, and at an exponential rate (note that $P(N>i)=(1-p)^i \leq e^{-pi}$).

The rejection method has an almost unlimited potential. We have given up the principle that one uniform [0,1] random variate yields one variate X (as in the inversion method), but what we receive in return is a powerful, simple and exact algorithm.

Example 3.1. Bounded densities of compact support.

Let $C_{M,a,b}$ be the class of all densities on $[a,b]$ bounded by M. Any such density is clearly bounded by M. Thus, the rejection algorithm can be used with uniform dominating density $g(x)=(b-a)^{-1}$ $(a \leq x \leq b)$, and the constant c becomes $M(b-a)$. Formally, we have

The rejection method for $C_{M,a,b}$

REPEAT
  Generate two independent uniform [0,1] random variates U and V.
  Set $X \leftarrow a+(b-a)V$.
UNTIL $UM \leq f(X)$
RETURN X ■

The reader should be warned here that this algorithm can be horribly inefficient, and that the choice of a constant dominating curve should be avoided except in a few cases.

3.2. Development of good rejection algorithms.

Generally speaking, g is chosen from a class of easy densities. This class includes the uniform density, triangular densities, and most densities that can be generated quickly by the inversion method. The situation usually dictates which densities are considered as "easy". There are two major techniques for determining c and g in the inequality $f \leq cg$: one could first study the form of f and apply one of many analytical devices for obtaining inequalities. Many of these are illustrated throughout this book (collecting them in a special chapter would have forced us to duplicate too much material). While this approach gives often

quick results (see Example 3.2 below), it is ad hoc, and depends a lot on the mathematical background and insight of the designer. In a second approach, which is also illustrated in this section, one starts with a family of dominating densities g and chooses the density within that class for which c is smallest. This approach is more structured but could sometimes lead to difficult optimization problems.

Example 3.2. A normal generator by rejection from the Laplace density.

Let f be the normal density. Obtaining an upper bound for f boils down to obtaining a lower bound for $\frac{x^2}{2}$. But we have of course

$$\frac{1}{2}(|x|-1)^2 = \frac{x^2}{2}+\frac{1}{2}-|x| \geq 0 .$$

Thus,

$$\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}} \leq \frac{1}{\sqrt{2\pi}}e^{\frac{1}{2}-|x|} = cg(x) ,$$

where $g(x) = \frac{1}{2}e^{-|x|}$ is the Laplace density, and $c = \sqrt{\frac{2e}{\pi}}$ is the rejection constant. This suggests the following algorithm:

A normal generator by the rejection method

REPEAT

Generate an exponential random variate X and two independent uniform [0,1] random variates U and V. If $U < \frac{1}{2}$, set $X \leftarrow -X$ (X is now distributed as a Laplace random variate).

UNTIL $V\frac{1}{\sqrt{2\pi}}e^{\frac{1}{2}-|X|} \leq \frac{1}{\sqrt{2\pi}}e^{-\frac{X^2}{2}}$.

RETURN X

The condition in the UNTIL statement can be cleaned up. The constant $\frac{1}{\sqrt{2\pi}}$ cancels out on left and right hand sides. It is also better to take logarithms on both sides. Finally, we can move the sign change to the RETURN statement because there is no need for a sign change of a random variate that will be rejected. The random variate U can also be avoided by the trick implemented in the algorithm given below.

A normal generator by rejection from the Laplace density

REPEAT

Generate an exponential random variate X and an independent uniform $[-1,1]$ random variate V.

UNTIL $(X-1)^2 \leq -2\log(|V|)$

RETURN $X \leftarrow X$ sign (V) ■

For given densities f and g, the rejection constant c should be at least equal to

$$\sup_x \frac{f(x)}{g(x)} .$$

We cannot loose anything by setting c equal to this supremum, because this insures us that the curves of f and cg touch each other somewhere. Instead of letting g be determined by some inequality which we happen to come across as in Example 3.2, it is often wiser to take the best g_θ in a family of densities parametrized by θ. Here θ should be thought of as a subset of R^k (in which case we say that there are k parameters). Define the optimal rejection constant by

$$c_\theta = \sup_x \frac{f(x)}{g_\theta(x)} .$$

The optimal θ is that for which c_θ is minimal, i.e. for which c_θ is closest to 1.

We will now illustrate this optimization process by an example. For the sake of argument, we take once again the normal density f. The family of dominating densities is the Cauchy family with scale parameter θ:

$$g_\theta(x) = \frac{\theta}{\pi} \frac{1}{\theta^2 + x^2} .$$

There is no need to consider a translation parameter as well because both f and the Cauchy densities are unimodal with peak at 0. Let us first compute the optimal rejection constant c_θ. We will prove that

$$c_\theta = \begin{cases} \frac{\sqrt{2\pi}}{e\,\theta} e^{\frac{\theta^2}{2}} & ,\theta < \sqrt{2} \\ \theta\sqrt{\frac{\pi}{2}} & ,\theta \geq \sqrt{2} \end{cases} .$$

We argue as follows: f/g_θ is maximal when $\log(f/g_\theta)$ is maximal. Setting the derivative with respect to x of $\log(f/g_\theta)$ equal to 0 yields the equation

$$-x+\frac{2x}{\theta^2+x^2}=0\ .$$

This gives the values $x=0$ and $x=\pm\sqrt{2-\theta^2}$ (the latter case can only happen when $\theta^2\leq 2$). At $x=0$, f/g_θ takes the value $\theta\sqrt{\frac{\pi}{2}}$. At $x=\pm\sqrt{2-\theta^2}$, f/g_θ takes the value $\frac{\sqrt{2\pi}}{e\,\theta}e^{\frac{\theta^2}{2}}$. It is easy to see that for $\theta<\sqrt{2}$, the maximum of f/g_θ is attained at $x=\pm\sqrt{2-\theta^2}$ and the minimum at $x=0$. For $\theta\geq\sqrt{2}$, the maximum is attained at $x=0$. This concludes the verification of the expression for c_θ.

The remainder of the optimization is simple. The function c_θ has only one minimum, at $\theta=1$. The minimal value is $c_1=\sqrt{\frac{2\pi}{e}}$. With this value, the condition of acceptance $Uc_\theta g_\theta(X)\leq f(X)$ can be rewritten as

$$U\sqrt{\frac{2\pi}{e}}\frac{1}{\pi}\frac{1}{1+X^2}\leq\frac{1}{\sqrt{2\pi}}e^{-\frac{X^2}{2}}\ ,$$

or as

$$U\leq(1+X^2)\frac{\sqrt{e}}{2}e^{-\frac{X^2}{2}}\ .$$

A normal generator by rejection from the Cauchy density

[SET-UP]
$\alpha\leftarrow\frac{\sqrt{e}}{2}$
[GENERATOR]
REPEAT
    Generate two independent uniform [0,1] random variates U and V.
    Set $X\leftarrow\tan(\pi V)$, $S\leftarrow X^2$ (X is now Cauchy distributed).
UNTIL $U\leq\alpha(1+S)e^{-\frac{S}{2}}$
RETURN X

The algorithm derived here, though it has a rejection constant near 1.4 is no match for most normal generators developed further on. The reason for this is that we need fairly expensive Cauchy random variates, plus the evaluation of exp in the acceptance step.

3.3. Generalizations of the rejection method.

Some generalizations of the rejection method are important enough to warrant special treatment in this key chapter. The first generalization concerns the following case:

$$f(x) = c\ g(x)\ \psi(x),$$

where the function ψ is [0,1]-valued, g is an easy density and c is a normalization constant at least equal to 1. The rejection algorithm for this case can be rewritten as follows:

The rejection method

REPEAT

Generate independent random variates X, U where X has density g and U is uniformly distributed on [0,1].

UNTIL $U \leq \psi(x)$

RETURN X

Vaduva (1977) observed that for special forms of ψ, there is another way of proceeding. This occurs when $\psi = 1-\Psi$ where Ψ is a distribution function of an easy density.

Vaduva's generalization of the rejection method

REPEAT

Generate two independent random variates X, Y, where X has density g and Y has distribution function Ψ.

UNTIL $X \leq Y$

RETURN X

For $\psi = \Psi$, we need to replace $X \leq Y$ in the acceptance step by $X \geq Y$.

Theorem 3.3.

Vaduva's rejection method produces a random variate X with density $f = cg(1-\Psi)$, and the rejection constant (the expected number of iterations) is c.

Proof of Theorem 3.3.

We prove this by showing that Vaduva's algorithm is entirely equivalent to the original rejection algorithm. Note that the condition of acceptance, $X \leq Y$ is with probability one satisfied if and only if $1-\Psi(X) \geq 1-\Psi(Y)$. But by the probability integral transform, we know that $1-\Psi(Y)$ is distributed as U, a uniform [0,1] random variable. Thus, we need only verify whether $U \leq 1-\Psi(X)$, which yields the original acceptance condition given at the beginning of this section. ■

The choice between generating U and computing $1-\Psi(X)$ on the one hand (the original rejection algorithm) and generating Y with distribution function Ψ on the other hand (Vaduva's method) depends mainly upon the relative speeds of computing a distribution function and generating a random variate with that distribution.

Example 3.3.

Consider the density

$$f(x) = c\ (ax^{a-1})\ e^{-x}\ ,0<x\leq 1\ ,$$

where $a>0$ is a parameter and c is a normalization constant. This density is part of the gamma (a) density, written here in a form convenient to us. The dominating density is $g(x)=ax^{a-1}$, and the function ψ is e^{-x}. Random variates with density g can be obtained quite easily by inversion (take $V^{\frac{1}{a}}$ where V is a uniform [0,1] random variate). In this case, the ordinary rejection algorithm would be

REPEAT

Generate two iid uniform [0,1] random variates U, V, and set $X \leftarrow V^{\frac{1}{a}}$.

UNTIL $U \leq e^{-X}$

RETURN X

Vaduva's modification essentially consists in generating X and an exponential random variate E until $E \geq X$. It is faster if we can generate E faster than we can compute e^{-X} (this is sometimes the case). Of course, in this simple example, we could have deduced Vaduva's modification by taking the logarithm of the acceptance condition and noting that E is distributed as $-\log(U)$. ■

We now proceed with another generalization found in Devroye (1984):

Theorem 3.4.

Assume that a density f on R^d can be decomposed as follows:

$$f(x) = \int g(y,x)\, h(y,x)\, dy \ ,$$

where $\int dy$ is an integral in R^k, $g(y,x)$ is a density in y for all x, and there exists a function $H(x)$ such that $0 \le h(y,x) \le H(x)$ for all y, and $H/\int H$ is an easy density. Then the following algorithm produces a random variate with density f, and takes N iterations where N is geometrically distributed with parameter $\frac{1}{\int H}$ (and thus $E(N) = \int H$).

Generalized rejection method

REPEAT
  Generate X with density $H/\int H$ (on R^d).
  Generate Y with density $g(y,X), y \in R^k$ (X is fixed).
  Generate a uniform [0,1] random variate U.
UNTIL $UH(X) \le h(Y,X)$
RETURN X

Proof of Theorem 3.4.

We will prove that this Theorem follows directly from Theorem 3.2. Let us define the new random vector $W_1 = (X,Y,U)$ where W_1 refers to the triple generated in the REPEAT loop. Then, if A is the set of values $w_1 = (x,y,u)$ for which $uH(x) \le h(y,x)$, we have for all Borel sets B in the space of w_1,

$$P(W \in B) = \frac{P(W_1 \in A \cap B)}{p}$$

where $p = P(W_1 \in A)$ and W refers to the value of W_1 upon exit. Take $B = (-\infty, x] \times R^k \times [0,1]$, and conclude that

$$P(X(\text{returned}) \le x) = \frac{1}{p} P(X \le x, UH(X) \le h(Y,X)) \ .$$

$$= \int_{-\infty}^{x} \int g(y,z)\, \frac{h(y,z)}{H(z)}\, \frac{H(z)}{\int H}\, dz\, dy$$

$$= \frac{1}{p \int H} \int_{-\infty}^{x} f(z)\, dz \ .$$

We note first that by setting $x = \infty$, $p = \frac{1}{\int H}$. But then, clearly, the variate produced by the algorithm has density f as required. ■

3.4. Wald's equation.

We will rather often be asked to evaluate the expected value of

$$\sum_{i=1}^{N} \psi(W_i) \ ,$$

where W_i is the collection of all random variables used in the i-th iteration of the rejection algorithm, ψ is some function, and N is the number of iterations of the rejection method. The random variable N is known as a stopping rule because the probabilities $P(N=n)$ are equal to the probabilities that $W_1, \ldots, W_n$ belong to some set B_n. The interesting fact is that, regardless of which stopping rule is used (i.e., whether we use the one suggested in the rejection method or not), as long as the W_i's are iid random variables, the following remains true:

Theorem 3.5. (Wald's equation.)

Assume that $W_1, \ldots$ are iid R^d-valued random variables, and that ψ is an arbitrary nonnegative Borel measurable function on R^d. Then, for all stopping rules N,

$$E\left(\sum_{i=1}^{N} \psi(W_i)\right) = E(N)\, E(\psi(W_1)) \ .$$

Proof of Theorem 3.5.

To simplify the notation we write $Z_i = \psi(W_i)$ and note that the Z_i's are iid nonnegative random variables. The proof given here is standard (see e.g. Chow and Teicher (1978, pp. 137-138)), but will be given in its entirety. We start by noting that Z_i and $I_{[N<i]}$ are independent for all i. Thus, so are Z_i and $I_{[N \geq i]}$. We will assume that $E(Z_1) < \infty$ and $E(N) < \infty$. It is easy to verify that the chain of equalities given below remains valid when one or both of these expectations is ∞.

$$E\left(\sum_{i=1}^{N} Z_i\right) = E\left(\sum_{i=1}^{\infty} Z_i I_{[N \geq i]}\right)$$

$$= \sum_{i=1}^{\infty} E(Z_i I_{[N \geq i]})$$

$$= \sum_{i=1}^{\infty} E(Z_i) P(N \geq i)$$

$$= E(Z_1) \sum_{i=1}^{\infty} P(N \geq i)$$

$$= E(Z_1)\, E(N)\,.$$

The exchange of the expectation and infinite sum is allowed by the monotone convergence theorem: just note that for any sequence of nonnegative random variables $Y_1, \ldots$, $\sum_{i=1}^{n} E(Y_i) = E(\sum_{i=1}^{n} Y_i) \to E(\sum_{i=1}^{\infty} Y_i)$. ■

It should be noted that for the rejection method, we have a special case for which a shorter proof can be given because our stopping rule N is an instantaneous stopping rule: we define a number of decisions D_i, all 0 or 1 valued and dependent upon W_i only: $D_1=0$ indicates that we "reject" based upon W_1, etcetera. A 1 denotes acceptance. Thus, N is equal to n if and only if $D_n=1$ and $D_i=0$ for all $i<n$. Now,

$$E(\sum_{i=1}^{N} \psi(W_i))$$

$$= E(\sum_{i<N} \psi(W_i)) + E(\psi(W_N))$$

$$= E(N-1)E(\psi(W_1) \mid D_1=0) + E(\psi(W_1) \mid D_1=1)$$

$$= (\frac{1}{P(D_1=1)} - 1)\frac{E(\psi(W_1)I_{D_1=0})}{P(D_1=0)} + \frac{E(\psi(W_1)I_{D_1=1})}{P(D_1=1)}$$

$$= \frac{E(\psi(W_1))}{P(D_1=1)}\,,$$

which proves this special case of Theorem 3.5.

3.5. Letac's lower bound.

In a profound but little publicized paper, Letac (1975) asks which distributions can be obtained for $X = U_N$ where N is a stopping time and $U_1, U_2, \ldots$ is an iid sequence of uniform [0,1] random variables. He shows among other things that all densities on [0,1] can be obtained in this manner. In exercise 3.14, one universal stopping time will be described. It does not coincide with Letac's universal stopping rule, but will do for didactical purposes.

More importantly, Letac has obtained lower bounds on the performance of any algorithm of this type. His main result is:

Theorem 3.6. (Letac's lower bound)

Assume that $X = U_N$ has density f on [0,1], where N and the U_i's are as defined above. For any such stopping rule N (i.e., for any algorithm), we have

$$E(N) \geq ||f||_\infty ,$$

where $||.||_\infty$ is the essential supremum of f.

Proof of Theorem 3.6.

By the independence of the events $[N \geq n]$ and $[U_n \in B]$ (which was also used in the proof of Wald's equation), we have

$$P(N \geq n, U_n \in B) = P(N \geq n) P(U_1 \in B) .$$

But,

$$\begin{aligned} P(X \in B) &= \sum_{n=1}^{\infty} P(N = n, U_n \in B) \\ &\leq \sum_{n=1}^{\infty} P(N \geq n, U_n \in B) \\ &= \sum_{n=1}^{\infty} P(N \geq n) P(U_1 \in B) \\ &= E(N) P(U_1 \in B) . \end{aligned}$$

Thus, for all Borel sets B,

$$E(N) \geq \frac{P(X \in B)}{P(U_1 \in B)} .$$

If we take the supremum of the right-hand-side over all B, then we obtain $||f||_\infty$. ■

There are quite a few algorithms that fall into this category. In particular, if we use rejection with a constant dominating curve on [0,1], then we use N uniform random variates where for continuous f ,

$$E(N) \geq \sup_x f(x) .$$

We have seen that in the rejection algorithm, we come within a factor of 2 of this lower bound. If the U_i's have density g on the real line, then we can construct stopping times for all densities f that are absolutely continuous with respect to g , and the lower bound reads

$$E(N) \geq \left|\left| \frac{f}{g} \right|\right|_\infty .$$

For continuous $\frac{f}{g}$, the lower bound is equal to $\sup \frac{f}{g}$ of course. Again, with the rejection method with g as dominating density, we come within a factor of 2 of the lower bound.

There is another class of algorithms that fits the description given here, notably the Forsythe-von Neumann algorithms, which will be presented in section IV.2.

3.6. The squeeze principle.

In the rejection method based on the inequality $f \leq cg$, we need to compute the ratio $\frac{f}{g}$ N times where N is the number of iterations. In most cases, this is a slow operation because f is presumably not a simple function of its argument (for otherwise, we would know how to generate random variates from f by other means). In fact, sometimes f is not known explicitly: in this book, we will encounter cases in which it is the integral of another function or the solution of a nonlinear equation. In all these situations, we should try to avoid the computation of $\frac{f}{g}$ either entirely, or at least most of the time. For principles leading to the total avoidance of the computation, we refer to the more advanced chapter IV. Here we will briefly discuss the squeeze principle (a term introduced by George Marsaglia (1977)) designed to avoid the computation of the ratio with high probability. One should in fact try to find functions h_1 and h_2 that are easy to evaluate and have the property that

$$h_1(x) \leq f(x) \leq h_2(x) .$$

Then, we have:

The squeeze method

REPEAT
  Generate a uniform [0,1] random variate U.
  Generate a random variate X with density g.
  Set $W \leftarrow Ucg(X)$.
  Accept $\leftarrow [W \leq h_1(X)]$.
  IF NOT Accept
    THEN IF $W \leq h_2(X)$ THEN Accept $\leftarrow [W \leq f(X)]$.
UNTIL Accept
RETURN X

In this algorithm, we introduced the boolean variable "Accept" to streamline the exit from the REPEAT loop. Such boolean variables come in handy whenever a program must remain structured and readable. In the algorithm, we count on the fact that "Accept" gets its value most of the time from the comparison between W and $h_1(X)$, which from now on will be called a quick acceptance step. In the remaining cases, we use a quick rejection step ($W > h_2(X)$), and in the rare cases that W is sandwiched between $h_1(X)$ and $h_2(X)$, we resort to the expensive comparison of W with $f(X)$ to set the value of "Accept".

The validity of the algorithm is not jeopardized by dropping the quick acceptance and quick rejection steps. In that case, we simply have the statement Accept$\leftarrow [W \leq f(X)]$, and obtain the standard rejection algorithm. In many cases, the quick rejection step is omitted since it has the smallest effect on the efficiency. Note also that it is not necessary that $h_1 \geq 0$ or $h_2 \leq cg$, although nothing will be gained by considering violations of these boundary conditions.

We note that N is as in the rejection algorithm, and thus, $E(N) = c$. The gain will be in the number of computations N_f of f, the dominating factor in the time complexity. The computation of $E(N_f)$ demonstrates the usefulness of Wald's equation once again. Indeed, we have

$$N_f = \sum_{i=1}^{N} I_{[h_1(X_i) < W_i < h_2(X_i)]},$$

where W_i is the W obtained in the i-th iteration, and X_i is the X used in the i-th iteration. To this sum, we can apply Wald's equation, and thus,

$$\begin{aligned} E(N_f) &= E(N)\, P(h_1(X_1) < W_1 < h_2(X_1)) \\ &= c \int g(x) \frac{h_2(x) - h_1(x)}{cg(x)}\, dx \\ &= \int (h_2(x) - h_1(x))\, dx \,. \end{aligned}$$

Here we used the fact that we have proper sandwiching, i.e. $0 \le h_1 \le f \le h_2 \le cg$. If $h_1 \equiv 0$ and $h_2 \equiv cg$ (i.e., we have no squeezing), then we obtain the result $E(N_f) = c$ for the rejection method. With only a quick acceptance step (i.e. $h_2 = cg$), we have $E(N_f) = c - \int h_1$. When $h_1 \ge 0$ and/or $h_2 \le cg$ are violated, equality in the expression for $E(N_f)$ should be replaced by inequality (exercise 3.13).

Inequalities via Taylor's series expansion.

A good source of inequalities for functions f in terms of simpler functions is provided by Taylor's series expansion. If f has n continuous derivatives (denoted by $f^{(1)}, \ldots, f^{(n)}$), then it is known that

$$f(x) = f(0) + \frac{x}{1!} f^{(1)}(0) + \cdots + \frac{x^{n-1}}{n-1!} f^{(n-1)}(0) + \frac{x^n}{n!} f^{(n)}(\xi),$$

where ξ is a number in the interval $[0,x]$ (or $[x,0]$, depending upon the sign of x). From this, by inspection of the last term, one can obtain inequalities which are polynomials, and thus prime candidates for h_1 and h_2. For example, we have

$$e^{-x} = 1 - x + \frac{x^2}{2!} - \cdots + (-1)^{n-1} \frac{x^{n-1}}{n-1!} + (-1)^n \frac{x^n}{n!} e^{-\xi}.$$

From this, we see that for $x \ge 0$, e^{-x} is sandwiched between consecutive terms of the well-known expansion

$$e^{-x} = \sum_{i=0}^{\infty} (-1)^i \frac{x^i}{i!}.$$

In particular,

$$1 - x \le e^{-x} \le 1 - x + \frac{x^2}{2} \qquad (x \ge 0). \blacksquare$$

Example 3.4. The normal density.

For the normal density f, we have developed an algorithm based upon rejection from the Cauchy density in Example 3.2. We used the inequality $f \le cg$ where $c = \sqrt{\frac{2\pi}{e}}$ and $g(x) = \frac{1}{\pi(1+x^2)}$. For h_1 and h_2 we should look for simple functions of x. Applying the Taylor series technique described above, we see that

$$1 - \frac{x^2}{2} \le \sqrt{2\pi} f(x) \le 1 - \frac{x^2}{2} + \frac{x^4}{8}.$$

Using the lower bound for h_1, we can now accelerate our normal random variate generator somewhat:

Normal variate generator by rejection and squeezing

REPEAT
  Generate a uniform [0,1] random variate U.
  Generate a Cauchy random variate X.
  Set $W \leftarrow \frac{2U}{\sqrt{e}\,(1+X^2)}$. (Note: $W \leftarrow cUg(X)\sqrt{2\pi}$.)
  Accept $\leftarrow [W \leq 1-\frac{X^2}{2}]$.
  IF NOT Accept THEN Accept $\leftarrow [W \leq e^{-\frac{X^2}{2}}]$.
UNTIL Accept
RETURN X

This algorithm can be improved in many directions. We have already got rid of the annoying normalization constant $\sqrt{2\pi}$. For $|X| > \sqrt{2}$, the quick acceptance step is useless in view of $h_1(X) < 0$. Some further savings in computer time result if we work with $Y \leftarrow \frac{1}{2}X^2$ throughout. The expected number of computations of f is

$$c - \int h_1 = \sqrt{\frac{2\pi}{e}} - \frac{1}{\sqrt{2\pi}} \int_{|x| \leq \sqrt{2}} (1-\frac{x^2}{2})\,dx = \sqrt{\frac{2\pi}{e}} - \frac{4}{3\sqrt{\pi}} \,. \blacksquare$$

Example 3.5. Proportional squeezing.

It is sometimes advantageous to sandwich f between two functions of the same form as in

$$bg \leq f \leq cg \,,$$

where g is an easy density (as in the rejection method), and b is a positive constant. When b and c are close to 1, such a proportional squeeze can be very useful. For example, random variates can be generated as follows:

The proportional squeeze method

REPEAT

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

Generate a random variate X with density g.

Accept $\leftarrow [U \leq \frac{b}{c}]$.

IF NOT Accept THEN Accept $\leftarrow [U \leq \frac{f(X)}{cg(X)}]$.

UNTIL Accept

RETURN X

Here the expected number of computations of f is quite simply $c-b$. The main area of application of this method is in the development of universally applicable algorithms in which the real line is first partitioned into many intervals. On each interval, we have a nearly constant or nearly linear piece of density. For this piece, proportional squeezing with dominating density of the form $g(x)=a_0+a_1x$ can usually be applied (see exercises 3.10 and 3.11 below). ■

Example 3.6. Squeezing based upon an absolute deviation inequality.

Assume that a density f is close to another density h in the following sense:

$$|f-h| \leq g .$$

Here g is another function, typically with small integral. Here we could implement the rejection method with as dominating curve $g+h$, and apply a sueeze step based upon $f \geq h-g$. After some simplifications, this leads to the following algorithm:

REPEAT

Generate a random variate X with density proportional to $h+g$, and a uniform [0,1] random variate U.

Accept $\leftarrow [\frac{g(X)}{h(X)} \leq \frac{1-U}{1+U}]$

IF NOT Accept THEN Accept $\leftarrow [U(g(X)+h(X)) \leq f(X)]$

UNTIL Accept

RETURN X

This algorithm has rejection constant $1+\int g$, and the expected number of evaluations of f is at most $2\int g$. Algorithms of this type are mainly used when g has very small integral. One instance is when the starting absolute deviation inequality is known from the study of limit theorems in mathematical statistics. For example, when f is the gamma (n) density normalized to have zero mean and unit variance, it is known that f tends to the normal density as $n \to \infty$. This convergence is studied in more detail in local central limit theorems (see e.g. Petrov (1975)). One of the by-products of this theory is an inequality of the form needed by us, where g is a function depending upon n, with integral decreasing at the rate $1/\sqrt{n}$ as $n \to \infty$. The rejection algorithm would thus have improved performance as $n \to \infty$. What is intriguing here is that this sort of inequality is not limited to the gamma density, but applies to densities of sums of iid random variables satisfying certain regularity conditions. In one sweep, one could thus design general algorithms for this class of densities. See also sections XIV.3.3 and XIV.4. ■

3.7. Recycling random variates.

In this section we have emphasized the expected number of iterations in our algorithms. Sometimes we have looked at the number of function evaluations. But by and large we have steered clear of making statements about the expected number of uniform random variates needed before an algorithm halts. One of the reasons is that we can always recycle unused parts of the uniform random variate. The recycling principle is harmless for our infinite precision model, but should be used with extreme care in standard single precision arithmetic on computers.

For the rejection method, based upon the inequality $f \leq cg$ where g is the dominating density, and c is a constant, we note that given a random variate X

with density g and an independent uniform [0,1] random variate U, the halting rule is $Ucg(X)/f(X) \leq 1$. Given that we halt, then $Ucg(X)/f(X)$ is again uniform on [0,1]. If we reject, then

$$\frac{\frac{Ucg(X)}{f(X)} - 1}{\frac{g(X)}{f(X)} - 1}$$

is again uniformly distributed on [0,1]. These recycled uniforms can be used either in the generation of the next random variate (if more than one random variate is needed), or in the next iteration of the rejection algorithm. Thus, in theory, the cost of uniform [0,1] random variates becomes negligible: it is one if only one random variate must be generated, and it remains one even if n random variates are needed. The following algorithm incorporates these ideas:

Rejection algorithm with recycling of one uniform random variate

Generate a uniform [0,1] random variate U.
REPEAT
  REPEAT
    Generate a random variate X with density g.
    $T \leftarrow \frac{cg(X)}{f(X)}$, $V \leftarrow UT$
    $U \leftarrow \frac{V-1}{T-1}$ (prepare for recycling)
  UNTIL $U \leq 0$ (equivalent to $V \leq 1$)
  RETURN X (X has density f)
  $U \leftarrow V$ (recycle)
UNTIL False (this is an infinite loop; add stopping rule)

In this example, we merely want to make a point about our idealized model. Recycling can be (and usually is) dangerous on finite-precision computers. When f is close to cg, as in most good rejection algorithms, the upper portion of U (i.e. $(V-1)/(T-1)$ in the notation of the algorithm) should not be recycled since $T-1$ is close to 0. The bottom part is more useful, but this is at the expense of less readable algorithms. All programs should be set up as follows: a uniform random variate should be provided upon input, and the output consists of the returned random variate and another uniform random variate. The input and output random variates are dependent, but it should be stressed that the returned random variate X and the recycled uniform random variate are independent! Another argument against recycling is that it requires a few multiplications and/or divisions. Typically, the time taken by these operations is longer than the time needed to generate one good uniform [0,1] random variate. For all these reasons, we do not pursue the recycling principle any further.

3.8. Exercises.

1. Let f and g be easy densities for which we have subprograms for computing $f(x)$ and $g(x)$ at all $x \in R^d$. These densities can be combined into other densities in several manners, e.g.

$$h = c \max(f, g)$$
$$h = c \min(f, g)$$
$$h = c \sqrt{fg}$$
$$h = c f^{\alpha} g^{1-\alpha}$$

where c is a normalization constant (different in each case) and $\alpha \in [0,1]$ is a constant. How would you generate random variates with density h ? Give the expected time complexity (expected number of iterations, comparisons, etc.).

2. Decompose the density $h(x) = \frac{2}{\pi}\sqrt{1-x^2}$ on $[-1,1]$ as follows:

$$h(x) = c \sqrt{f(x)\, g(x)}$$

where $c = \frac{2}{\pi}\sqrt{\frac{8}{3}}$, $f(x) = \frac{3}{4}(1-x^2)$ and $g(x) = \frac{1}{2}$, and $|x| \leq 1$. Thus, h is in one of the forms specified in exercise 3.1. Give a complete algorithm and analysis for generating random variates with density h by the general method of exercise 3.1.

3. The algorithm

```
REPEAT
    Generate X with density g.
    Generate an exponential random variate E.
UNTIL h(X) ≤ E
RETURN X
```

when used with a nonnegative function h produces a random variate X with density

$$c\, g(x)\, e^{-h(x)},$$

where c is a normalization constant. Show this.

4. How does c, the rejection constant, change with n (i.e., what is its rate of increase as $n \to \infty$) when the rejection method is used on the beta (n, n) density and the dominating density g is the uniform density on $[0,1]$?

5. Lux (1979) has generalized the rejection method as follows. Let g be a given density, and let F be a given distribution function. Furthermore, assume

that r is a fixed positive-valued monotonically decreasing function on $[0,\infty)$. Then a random variate X with density

$$f(x) = g(x) \int_{-\infty}^{r(x)} \left(\frac{1}{\int_0^{r^{-1}(y)} g(z)\,dz} \right) dF(y) \qquad (x>0):$$

Lux's algorithm

REPEAT
  Generate a random variate X with density g.
  Generate a random variate Y with distribution function F.
UNTIL $Y \le r(X)$
RETURN X

Also, the probability of acceptance of a random couple (X,Y) in Lux's algorithm is $\int_0^\infty F(r(x))\,g(x)\,dx$.

6. The following density on $[0,\infty)$ has both an infinite peak at 0 and a heavy tail:

$$f(x) = \frac{2}{(1+x)\sqrt{x^2+2x}} \qquad (x>0).$$

Consider as a possible candidate for a dominating curve $c_\theta\, g_\theta$ where

$$c_\theta\, g_\theta(x) = \begin{cases} \dfrac{2}{\pi\sqrt{2x}} & ,0 \le x \le \theta \\ \dfrac{2}{\pi x^2} & ,x>\theta \end{cases},$$

where c_θ is a constant depending upon θ only and $\theta>0$ is a design parameter. Prove first that indeed $f \le c_\theta\, g_\theta$. Then show that c_θ is minimal for $\theta=2^{1/3}$ and takes the value $\dfrac{6}{\pi 2^{\frac{1}{3}}}$. Give also a description of the entire rejection algorithm together with the values for the expected number of iterations, comparisons, square root operations, multiplications/divisions, and assignment statements. Repeat the same exercise when the dominating density is the density of the random variable $\theta U^2/V$ where $\theta>0$ is a parameter and U and V are two iid uniform $[0,1]$ random variates. Prove that in this case too we obtain the same rejection constant $\dfrac{6}{\pi 2^{\frac{1}{3}}}$.

7. **Optimal rejection algorithms for the normal density.** Assume that normal random variates are generated by rejection from a density g_θ where θ is a design parameter. Depending upon the class of g_θ's that is considered, we may obtain different optimal rejection constants. Complete the following table:

$g_\theta(x)$	Optimal θ	Optimal rejection constant c
Cauchy (θ): $\frac{\theta}{\pi(\theta^2+x^2)}$	1	$\sqrt{\frac{2\pi}{e}}$
Laplace (θ): $\frac{\theta}{2}e^{-\theta\|x\|}$	1	$\sqrt{\frac{2e}{\pi}}$
Logistic (θ): $\frac{\theta e^{-\theta x}}{(1+e^{-\theta x})^2}$	?	?
$\min(\frac{1}{4\theta}, \frac{\theta}{4x^2})$	?	?

8. **Sibuya's modified rejection method.** Sibuya (1962) noted that the number of uniform random variates in the rejection algorithm can be reduced to one by repeated use of the same uniform random variate. His algorithm for generating a random variate with density f (known not to exceed cg for an easy density g) is:

Generate a uniform [0,1] random variate U.
REPEAT
    Generate a random variate X with density g.
UNTIL $cg(X)U \le f(X)$
RETURN X

Show the following:

(i) The algorithm is valid if and only if $c = \text{ess sup}\,(f(X)/g(X))$.

(ii) If N is the number of X's needed in Sibuya's algorithm, and N^* is the number of X's needed in the original rejection algorithm, then

$$E(N) \ge E(N^*)$$

and

$$P(N \ge i) \ge P(N^* \ge i) \quad (\text{all } i).$$

(Hint: use Jensen's inequality.) We conclude from (ii) that Sibuya's method is worse than the rejection method in terms of number of required iterations.

(iii) We can have $P(N=\infty)>0$ (just take $g=f$, $c>1$). We can also have $P(N=\infty)=0, E(N)=\infty$ (just take $f(x)=2(1-x)$ on [0,1], $c=2$

and $g(x)=1$ on $[0,1]$). Give a necessary and sufficient condition for $P(N=\infty)=0$, and show that this requires that c is chosen optimally.

See also Greenwood (1976).

9. There exists a second moment analog of Wald's equation which you should try to prove. Let $W_1, \ldots,$ and $\psi \geq 0$ be as in Theorem 3.5. Assume further that $\psi(W_1)$ has mean μ and variance $\sigma^2 < \infty$. Then, for any stopping rule N with $E(N) < \infty$,

$$E\left(\left(\sum_{i=1}^{N}(W_i-\mu)\right)^2\right) = \sigma^2 E(N) .$$

See for example Chow and Teicher (1978, pp. 139).

10. Assume that we use proportional squeezing for a density f on $[0,1]$ which is known to be between $2b(1-x)$ and $2c(1-x)$ where $0 \leq b \leq 1 \leq c < \infty$. Then, we need in every iteration a uniform random variate U and a triangular random variate X (which in turn can be obtained as $\min(U_1,U_2)$ where U_1,U_2 are also uniform $[0,1]$ random variates). Prove that if $U_{(1)} \leq U_{(2)}$ are the order statistics of U_1,U_2, then

$$\left(U_{(1)}, \frac{U_{(2)}-U_{(1)}}{1-U_{(1)}}\right)$$

is distributed as (X,U). Thus, using this device, we can "save" one uniform random variate per iteration. Write out the details of the corresponding proportional squeeze algorithm.

11. Assume that the density f has support on $[0,1]$ and that we know that it is Lipschitz with constant C, i.e.

$$|f(y)-f(x)| \leq C|x-y| \qquad (x,y \in R) .$$

Clearly, we have $f(0)=f(1)=0$. Give an efficient algorithm for generating a random variate with density f which is based upon an n-part equi-spaced partition of $[0,1]$ and the use of the proportional squeeze method for nearly linear densities (see previous exercise) for generating random variates from the n individual pieces. Your algorithm should be asymptotically efficient, i.e. it should have $E(N_f)=o(1)$ as $n \to \infty$ where N_f is the number of computations of f.

12. **Random variates with density $f(x)=c(1-x^2)^a$ ($|x| \leq 1$).** The family of densities treated in this exercise coincides with the family of symmetric beta densities properly translated and rescaled. For example, when the parameter a is integer, f is the density of the median of $2a+1$ iid uniform $[-1,1]$ random variates. It is also the density of the marginal distribution of a random vector uniformly distributed on the surface of the unit sphere in R^d where d and a are related by $a=\frac{d-3}{2}$. For the latter reason, we will use it later as an important tool in the generation of random vectors that are uniformly distributed on such spheres. The parameter a must be greater than -1. We have

$$c = \frac{\Gamma(a+\frac{3}{2})}{\sqrt{\pi}\Gamma(a+1)},$$

and the inequalities

$$ce^{-\frac{ax^2}{1-x^2}} \leq f(x) \leq ce^{-ax^2} \quad (|x| \leq 1).$$

The following rejection algorithm with squeezing can be used:

Translated symmetric beta generator by rejection and squeezing

REPEAT
  REPEAT
    Generate a normal random variate X.
    Generate an exponential random variate E.
  UNTIL $Y \leq 1$
  $X \leftarrow \frac{X}{\sqrt{2a}}$, $Y \leftarrow X^2$
  Accept $\leftarrow [1-Y(1+\frac{a}{E}Y) \geq 0]$.
  IF NOT Accept THEN Accept $\leftarrow [aY+E+a\log(1-Y) \geq 0]$.
UNTIL Accept
RETURN X

A. Verify that the algorithm is valid.

B. The expected number of normal/exponential pairs needed is $\frac{\Gamma(a+\frac{3}{2})}{\sqrt{a}\,\Gamma(a+1)}$. Selected values are

$a=1$	$\frac{3}{4}\sqrt{\pi}$	1.329340...
$a=2$	$\frac{15}{16}\sqrt{\frac{\pi}{2}}$	1.174982...
$a=3$	$\frac{105}{96}\sqrt{\frac{\pi}{3}}$	1.119263...

Show that this number tends to 1 as $a \to \infty$ and to ∞ as $a \downarrow 0$.

C. From part B we conclude that it is better to take care of the case $0 \leq a \leq 1$ separately, by bounding as follows: $c(1-x^2) \leq f(x) \leq c$. The expected number of iterations becomes $2c$, which takes the values $\frac{3}{2}$ at $a=1$ and 1 at $a=0$. Does this number vary monotonically with a? How does $E(N_f)$ vary with a?

D. Write a generator which works for all $a > -1$. (This requires yet another solution for a in the range $(-1,0)$.)

E. Random variates from f can also obtained in other ways. Show that all of the following recipes are valid:

(i) $S\sqrt{B}$ where B is beta$(\frac{1}{2}, a+1)$ and S is a random sign.

(ii) $S\sqrt{\frac{Y}{Y+Z}}$ where Y, Z are independent gamma$(\frac{1}{2},1)$ and gamma$(a+1,1)$ random variates, and S is a random sign.

(iii) $2B-1$ where B is a beta$(a+1, a+1)$ random variate.

13. Consider the squeeze algorithm of section 3.6 which uses the inequality $f \le cg$ for the rejection-based generator, and the inequalities $h_1 \le f \le h_2$ for the quick acceptance and rejection steps. Even if h_1 is not necessarily positive, and h_2 is not necessarily smaller than cg, show that we always have

$$E(N_f) = \int (\min(h_2, cg) - \max(h_1, 0)) \le \int (h_2 - h_1),$$

where N_f is the number of evaluations of f.

14. **A universal generator a la Letac.** Let f be **any** density on $[0,1]$, and assume that the cumulative mass function $M(t) = \int_{f \ge t} f(x)\,dx$ is known. Consider the following algorithm:

```
Generate a random integer Z where P(Z=i)=M(i)-M(i+1).
REPEAT
    Generate (X,V) uniformly in [0,1]^2
UNTIL Z+V ≤ f(X)
RETURN X
```

Show that the algorithm is valid (relate it to the rejection method). Relate the expected number of X's generated before halting to $||f||_\infty$, the essential supremum of f. Among other things, conclude that the expected time is ∞ for every unbounded density. Compare the expected number of X's with Letac's lower bound. Show also that if inversion by sequential search is used for generating Z, then the expected number of iterations in the search before halting is finite if and only if $\int f^2 < \infty$. A final note: usually, one does not have a cumulative mass function for an arbitrary density f.

4. DECOMPOSITION AS DISCRETE MIXTURES.

4.1. Definition.

If our target density f can be decomposed into a discrete mixture

$$f(x) = \sum_{i=1}^{\infty} p_i f_i(x)$$

where the f_i's are given densities and the p_i's form a probability vector (i.e., $p_i \geq 0$ for all i and $\sum_i p_i = 1$), then random variates can be obtained as follows:

The composition method.

Generate a random integer Z with probability vector $p_1, \ldots, p_i, \ldots$ (i.e. $P(Z=i)=p_i$).
Generate a random variate X with density f_Z.
RETURN X

This algorithm is incomplete, because it does not specify just how Z and X are generated. Every time we use the general form of the algorithm, we will say that the composition method is used.

We will show in this section how the decomposition method can be applied in the design of good generators, but we will not at this stage address the problem of the generation of the discrete random variate Z. Rather, we are interested in the decomposition itself. It should be noted however that in many, if not most, practical situations, we have a finite mixture with K components.

4.2. Decomposition into simple components.

Very often, we will decompose the graph of f into a bunch of very simple structures such as rectangles and triangles, mainly because random variates with rectangular-shaped or triangular-shaped densities are so easy to generate (by linear combinations of one or two uniform [0,1] random variates). This decomposition is finite if f is piecewise linear with a finite number of pieces (this forces f to have compact support). In general, one will decompose f as follows:

$$f(x) = \sum_{i=1}^{K-2} p_i f_i(x) + p_{K-1} f_{K-1}(x) + p_K f_K(x)$$

where f_K is a tail density (it is zero on a central interval $[a,b]$), p_K is usually very small, and all other f_i's vanish outside the central interval $[a,b]$. The structure of $f_1, \ldots, f_{K-2}$ is simple, e.g. rectangular. After having picked the rectangles in such a way that the corresponding p_i's add up to nearly 1, we

collect the leftover piece in $p_{K-1}f_{K-1}$. This last piece is often strangely shaped, and random variates from it are generated by the rejection method. The point is that p_{K-1} and p_K are so small that we do not have to generate random variates with this density very often. Most of the time, i.e. with probability $p_1+\cdots+p_{K-2}$, it suffices to generate one or two uniform [0,1] random variates and to shift or rescale them. This technique will be called the **jigsaw puzzle method**, a term coined by Marsaglia. The careful decomposition requires some refined analysis, and is usually only worth the trouble for frequently used fixed densities such as the normal density. We refer to the section on normal variate generation for several applications of this sort of decomposition. Occasionally, it can be applied to families of distributions (such as the beta and gamma families), but the problem is that the decomposition itself is a function of the parameter(s) of the family. This will be illustrated for the beta family (see section IX.4).

To give the readers a flavor of the sort of work that is involved, we will try to decompose the normal density into a rectangle and one residual piece: the rectangle will be called $p_1 f_1(x)$, and the residual piece $p_2 f_2(x)$. It is clear that p_1 should be as large as possible. But since $p_1 f_1(x) \le f(x)$, the largest p_1 must satisfy

$$p_1 \le \inf_x \frac{f(x)}{f_1(x)} .$$

Thus, with $f_1(x)=\frac{1}{2}\theta$, $|x| \le \theta$ where θ is the width of the centered rectangle, we see that at best we can set

$$p_1 = \inf_{|x| \le \theta} \frac{2\theta e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} = 2\frac{\theta}{\sqrt{2\pi}} e^{-\frac{\theta^2}{2}} .$$

The function p_1 is maximal (as a function of θ) when $\theta=1$, and the corresponding value is $\sqrt{\frac{2}{\pi e}}$. Of course, this weight is not close to 1, and the present decomposition seems hardly useful. The work involved when we decompose in terms of several rectangles and triangles is basically not different from the short analysis done here.

4.3. Partitions into intervals.

Many algorithms are based on the following principle: partition the real line into intervals $A_1, \ldots, A_K$, and decompose f as

$$f(x) = \sum_{i=1}^{K} f(x) I_{A_i}(x) .$$

If we can generate random variates from the restricted densities $f I_{A_i}/p_i$ (where $p_i = \int_{A_i} f$), then the decomposition method is applicable. The advantages offered

by partitions into intervals cannot be denied: the decomposition is so simple that it can be mechanized and used for huge classes of densities (in that case, there are usually very many intervals); troublespots on the real line such as infinite tails or unbounded peaks can be conveniently isolated; and most importantly, the decomposition is easily understood by the general user.

In some cases, random variates from the component densities are generated by the rejection method based on the inequalities

$$f(x) \leq h_i(x), x \in A_i, 1 \leq i \leq K .$$

Here the h_i's are given dominating curves. There are two subtly different methods for generating random variates with density f, given below. One of these needs the constants $p_i = \int_{A_i} f$, and the other one requires the constants $q_i = \int_{A_i} h_i$. Note that the q_i's are nearly always known because the h_i's are chosen by the user. The p_i's are usually known when the distribution function for F is easy to compute at arbitrary points.

The composition method.

Generate a discrete random variate Z with probability vector $p_1, \ldots, p_K$ on $\{1, \ldots, K\}$.
REPEAT
    Generate a random variate X with density h_i/q_i on A_i.
    Generate an independent uniform [0,1] random variate U.
UNTIL $Uh_i(X) \leq f(X)$
RETURN X

The modified composition method.

REPEAT

Generate a discrete random variate Z with probability vector proportional to $q_1, \ldots, q_K$ on $\{1, \ldots, K\}$.

Generate a random variate X with density h_i/q_i on A_i.

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

UNTIL $Uh_i(X) \leq f(X)$

RETURN X

In the second algorithm we use the rejection method with as dominating curve $h_1 I_{A_1} + \cdots + h_K I_{A_K}$, and use the composition method for random variates from the dominating density. In contrast, the first algorithm uses true decomposition. After having selected a component with the correct probability we then use the rejection method. A brief comparison of both algorithms is in order here. This can be done in terms of four quantities: N_Z, N_U, N_h and N_{h_i}, where N is the number of random variates required of the type specified by the index with the understanding that N_h refers to $\sum_{i=1}^{K} h_i$, i.e. it is the total number of random variates needed from any one of the K dominating densities.

Theorem 4.1.

Let $q = \sum_{i=1}^{K} q_i$, and let N be the number of iterations in the second algorithm. For the second algorithm we have $N_U = N_Z = N_h = N$, and N is geometrically distributed with parameter $\frac{1}{q}$. In particular,

$$E(N) = q \; ; \; E(N^2) = 2q^2 - q \; .$$

For the first algorithm, we have $N_Z = 1$. Also, $N_U = N_h$ satisfy

$$E(N_U) = q \; ; \; E(N_U{}^2) = \sum_{i=1}^{K} \frac{2q_i{}^2}{p_i} - q \geq 2q^2 - q \; .$$

Finally, for both algorithms, $E(N_{h_i}) = q_i$.

Proof of Theorem 4.1.

The statement for the second algorithm is obvious when we note that the rejection constant is equal to the area q under the dominating curve (the sum of the h_i's in this case). For the first algorithm, we observe that given the value of Z, N_U is geometrically distributed with parameter p_Z/q_Z. From the properties of the geometric distribution, we then conclude the following:

$$E(N_U) = \sum_{i=1}^{K} p_i(\frac{q_i}{p_i}) = \sum_{i=1}^{K} q_i = q \ ,$$

$$E(N_U{}^2) = \sum_{i=1}^{K} p_i(\frac{2}{(\frac{p_i}{q_i})^2} - \frac{1}{\frac{p_i}{q_i}}) = \sum_{i=1}^{K} 2p_i(\frac{q_i}{p_i})^2 - q \ .$$

To show that the last expression is always greater or equal to $2q^2-q$ we use the Cauchy-Schwarz inequality:

$$\sum_{i=1}^{K} 2p_i(\frac{q_i}{p_i})^2 - q \geq 2(\sum_{i=1}^{K} p_i \frac{q_i}{p_i})^2 (\sum_{i=1}^{K} p_i \times 1)^2 - q = 2q^2 - q \ .$$

Finally, we consider $E(N_{h_i})$. For the first algorithm, its expected value is $p_i(\frac{q_i}{p_i}) = q_i$. For the second algorithm, we employ Wald's equality after noting that $N_{h_i} = \sum_{j=1}^{N} I_{[\text{piece } h_i \text{ is used in the } j\text{-th iteration}]}$. Thus, the expected value is $E(N)P(\text{piece } h_i \text{ is used in the first iteration})$, which is equal to $q\ (\frac{q_i}{q}) = q_i$. ■

In standard circumstances, q is close to 1, and discrete random variate generators are ultra efficient. Thus, N_Z is not a great factor. For all the other quantities involved in the comparison, the expected values are equal. But when we examine the higher moments of the distributions, we notice a striking difference, because the second method has in all cases a smaller second moment. In fact, the difference can be substantial when for some i, the ratio q_i/p_i is large. If we take $q_i = p_i$ for $i \geq 2$ and $q_1 = q-(1-p_1)$, then for the first method,

$$E(N_U{}^2) = \frac{2(q-1+p_1)^2}{p_1} + 2(1-p_1) - q = (2q^2-q) + 2(q-1)^2(\frac{1}{p_1}-1) \ .$$

The difference between the two second moments in this example is $2(q-1)^2(\frac{1}{p_1}-1)$. Thus, isolating a small probability piece in the decomposition method and using a poor rejection rate for that particular piece is dangerous. In such situations, one is better off using a global rejection method as suggested in the second algorithm.

4.4. The waiting time method for asymmetric mixtures.

In large simulations, one needs iid random variates $X_1, \ldots, X_n, \ldots$. If these random variates are generated by the composition method, then for every random variate generated we need one discrete random variate Z for selecting a component. When f is decomposed into a main component $p_1 f_1$ (p_1 is close to 1) and a small component $p_2 f_2$, then most of these selections will choose the first component. In those cases, it is useful to generate the times of occurrence of selection of the second component instead. If the second component is selected at times $T_1, T_2, \ldots$, then it is not difficult to see that $T_1, T_2 - T_1, \ldots$ are iid geometric random variables with parameter p_2, i.e.

$$P(T_1 = i) = (1-p_2)^{i-1} p_2 \quad (i \geq 1) .$$

A random variate T_1 can be generated as $\left\lceil -\frac{E}{\log(p_2)} \right\rceil$ where E is an exponential random variate. Of course, we need to keep track of these times as we go along, occasionally generating a new time. These times need to be stored locally in subprograms for otherwise we need to pass them as parameters. In some cases, the overhead associated with passing an extra parameter is comparable to the time needed to generate a uniform random variate. Thus, one should carefully look at how the large simulation can be organized before using the geometric waiting times.

4.5. Polynomial densities on [0,1].

In this section, we consider densities of the form

$$f(x) = \sum_{i=0}^{K} c_i x^i \qquad (0 \leq x \leq 1) ,$$

where the c_i's are constants and K is a positive integer. Densities with polynomial forms are important further on as building blocks for constructing piecewise polynomial approximations of more general densities. If K is 0 or 1, we have the uniform and triangular densities, and random variate generation is no problem. There is also no problem when the c_i's are all nonnegative. To see this, we observe that the distribution function F is a mixture of the form

$$F(x) = \sum_{i=1}^{K+1} \left(\frac{c_{i-1}}{i}\right) x^i$$

where of course $\sum_{i=1}^{K+1} \frac{c_{i-1}}{i} = 1$. Since x^i is the distribution function of the maximum of i iid uniform [0,1] random variables, we can proceed as follows:

Generate a discrete random variate Z where $P(Z=i)=\frac{c_{i-1}}{i}$, $1 \le i \le K+1$.

RETURN X where X is generated as $\max(U_1, \ldots, U_Z)$ and the U_i's are iid uniform [0,1] random variates.

We have a nontrivial problem on our hands when one or more of the c_i's are negative. The solution given here is due to Ahrens and Dieter (1974), and can be applied whenever $c_0 + \sum_{i:c_i<0} c_i \ge 0$. They decompose f as follows: let A be the collection of integers in $\{0, \ldots, K\}$ for which $c_i \ge 0$, and let B the collection of indices in $\{0, \ldots, K\}$ for which $c_i < 0$. Then, we have

$$f(x) = \sum_{i=0}^{K} c_i x^i$$

$$= p_0 + \sum_{i \in A} \frac{c_i}{i+1}((i+1)x^i) + \sum_{i \in B}(-\frac{ic_i}{i+1})(\frac{i+1}{i}(1-x^i)) \qquad (0 \le x \le 1),$$

where $p_0 = c_0 + \sum_{i \in B} c_i$ (which is ≥ 0 by assumption). If we set p_i equal to $c_i/(i+1)$ for $i \in A$, $i \ge 1$, and to $-ic_i/(i+1)$ for $i \in B$, then $p_0, p_1, \ldots, p_K$ is a probability vector, and we have thus decomposed f as a finite mixture. Let us briefly mention how random variate generation for the component densities can be done.

Lemma 4.1.

Let $U_1, U_2, \ldots$ be iid uniform [0,1] random variables.

A. For $a > 1$, $U_1^{\frac{1}{a}} U_2$ has density

$$\frac{a}{a-1}(1-x^{a-1}) \qquad (0 \le x \le 1).$$

B. Let L be the index of the first U_i not equal to $\max(U_1, \ldots, U_n)$ for $n \ge 2$. Then U_L has density

$$\frac{n}{n-1}(1-x^{n-1}) \qquad (0 \le x \le 1).$$

C. The density of $\max(U_1, \ldots, U_n)$ is nx^{n-1} $(0 \le x \le 1)$.

Proof of Lemma 4.1.

Part C is trivially true. Part A is a good exercise on transformations of random variables. Part B has a particularly elegant short proof. The density of a randomly chosen U_i is 1 (all densities are understood to be on [0,1]). Thus, when f is the density of U_L, we must have

$$\frac{n-1}{n} f(x) + \frac{1}{n} n x^{n-1} = 1 .$$

This uses the fact that with probability $\frac{1}{n}$, the randomly chosen U_i is the maximal U_i, and that with the complimentary probability, the randomly chosen U_i is distributed as U_L. ■

We are now in a position to give more details of the polynomial density algorithm of Ahrens and Dieter.

Polynomial density algorithm of Ahrens and Dieter

[SET-UP]

Compute the probability vector $p_0, p_1, \ldots, p_K$ from $c_0, \ldots, c_K$ according to the formulas given above. For each $i \in \{0,1, \ldots, K\}$, store the membership of i ($i \in A$ if $c_i \geq 0$ and $i \in B$ otherwise).

[GENERATOR]

Generate a discrete random variate Z with probability vector $p_0, p_1, \ldots, p_K$.

IF $Z \in A$

THEN RETURN $X \leftarrow U^{\frac{1}{Z+1}}$ (or $X \leftarrow \max(U_1, \ldots, U_{Z+1})$ where $U, U_1, \ldots$ are iid uniform [0,1] random variates).

ELSE RETURN $X \leftarrow U_1^{\frac{1}{Z+1}} U_2$ (or $X \leftarrow U_L$ where L is the U_i with the lowest index not equal to $\max(U_1, \ldots, U_{Z+1})$).

4.6. Mixtures with negative coefficients.

Assume that the density $f(x)$ can be written as

$$f(x) = \sum_{i=1}^{\infty} p_i f_i(x),$$

where the f_i's are densities, but the p_i's are real numbers summing to one. A general algorithm for these densities was given by Bignami and de Matteis (1971). It uses the fact that if p_i is decomposed into its positive and negative parts, $p_i = p_{i+} - p_{i-}$, then

$$f(x) \leq g(x) = \sum_{i=1}^{\infty} p_{i+} f_i(x).$$

Then, the following rejection algorithm can be used:

Negative mixture algorithm of Bignami and de Matteis

REPEAT

Generate a random variate X with density $\sum_{i=1}^{\infty} p_{i+} f_i / \sum_{i=1}^{\infty} p_{i+}$.

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

UNTIL $U \sum_{i=1}^{\infty} p_{i+} f_i(X) \leq \sum_{i=1}^{\infty} p_i f_i(X)$

RETURN X

The rejection constant here is $\int g = \sum_{i=1}^{\infty} p_{i+}$. The algorithm is thus not valid when this constant is ∞. One should observe that for this algorithm, the rejection constant is probably not a good measure of the expected time taken by it. This is due to the fact that the time needed to verify the acceptance condition can be very large. For finite mixtures, or mixtures that are such that for every x, only a finite number of $f_i(x)$'s are nonzero, we are in good shape. In all cases, it is often possible to accept or reject after having computed just a few terms in the series, provided that we have good analytical estimates of the tail sums of the series. Since this is the main idea of the series method of section IV.5, it will not be pursued here any further.

Example 4.1.

The density $f(x) = \frac{3}{4}(1-x^2)$, $|x| \leq 1$, can be written as $f(x) = \frac{6}{4}(\frac{1}{2} I_{[-1,1]}(x)) - \frac{2}{4}(\frac{x^2}{6} I_{[-1,1]}(x))$. The algorithm given above is then

entirely equivalent to ordinary rejection from a uniform density, which in this case has a rejection constant of $\frac{3}{2}$:

REPEAT
  Generate a uniform [-1,1] random variate X.
  Generate a uniform [0,1] random variate U.
UNTIL $U \leq 1-X^2$
RETURN X ■

5. THE ACCEPTANCE-COMPLEMENT METHOD.

5.1. Definition.

Let f be a given density on R^d which can be decomposed into a sum of two nonnegative functions:

$$f(x) = f_1(x) + f_2(x).$$

Assume furthermore that there exists an easy density g such that $f_1 \leq g$. Then the following algorithm can be used to generate a random variate X with density f:

The acceptance-complement method

Generate a random variate X with density g.
Generate a uniform [0,1] random variate U.
IF $U > \frac{f_1(X)}{g(X)}$
  THEN Generate a random variate X with density $\frac{f_2}{p}$ where $p = \int f_2$.
RETURN X

This, the acceptance-complement method, was first proposed by Kronmal and Peterson (1981,1984). It generalizes the composition method as can be seen if we take $f_1 = f I_A$, $g = f_1 / \int f_1$ and $f_2 = f I_{A^c}$ where A is an arbitrary set of R^d

and A^c is its complement. It is competitive if three conditions are met:

(i) g is an easy density.

(ii) f_2/p is an easy density when p is not small (when p is small, this does not matter much).

(iii) f_1/g is not difficult to evaluate.

As with the composition method, the algorithm given above is more a principle than a detailed recipe. When we compare it with the rejection method, we notice that instead of one design variable (a dominating density) we find two design variables, f_2 and g. Moreover, there is no rejection involved at all, although very often, it turns out that a random variate from $\frac{f_2}{p}$ is generated by the rejection method.

Let us first show that this method is valid. For this purpose, we need only show that for all Borel sets $B \subseteq R^d$, the random variate generated by the algorithm (which will be denoted here by X) satisfies $P(X \in B) = \int_B f(x)\,dx$. To avoid confusion with too many X's, we will use Y for the random variate with density g. Thus,

$$P(X \in B) = P(Y \in B, U \le \frac{f_1(Y)}{g(Y)}) + P(U > \frac{f_1(Y)}{g(Y)}) \frac{\int_B f_2(x)\,dx}{p}$$

$$= \int_B g(x) \frac{f_1(x)}{g(x)}\,dx + (1 - \int g(x) \frac{f_1(x)}{g(x)}\,dx) \frac{\int_B f_2(x)\,dx}{p}$$

$$= \int_B f_1(x)\,dx + \int_B f_2(x)\,dx$$

$$= \int_B f(x)\,dx \ .$$

In general, we gain if we can RETURN the first X generated in the algorithm. Thus, it seems that we should try to maximize its probability of acceptance,

$$P(U \le \frac{f_1(Y)}{g(Y)}) = \int f_1 = 1 - p$$

subject of course to the constraint $f_1 \le g$ where g is an easy density. Thus, good algorithms have g "almost" equal to f.

There is a visual explanation of the method related to that of the rejection method. What is important here is that the areas under the graphs of $g - f_1$ and f_2 are equal. In the next section, we will give a simplified version of the acceptance-complement algorithm developed independently by Ahrens and Dieter (1981,1983). Examples and details are given in the remaining sections and in some of the exercises.

5.2. Simple acceptance-complement methods.

Ahrens and Dieter (1981,1983) and Deak (1981) considered the special case defined by an arbitrary density g on R^d and the following decomposition:

$$f(x) = f_1(x) + f_2(x);$$

$$f_1(x) = \min(f(x), g(x)) \quad (\text{note } : f_1 \leq g);$$

$$f_2(x) = (f(x) - g(x))_+ .$$

We can now rewrite the acceptance-complement algorithm quite simply as follows:

Simple acceptance-complement method of Ahrens and Dieter

Generate a random variate X with density g.
Generate a uniform [0,1] random variate U.
IF $U > \dfrac{f(X)}{g(X)}$
THEN Generate a random variate X with density $(f-g)_+/p$ where $p = \int_{f>g} (f-g)$.
RETURN X

Deak (1981) calls this the **economical method.** Usually, g is an easy density close to f. It should be obvious that generation from the leftover density $(f-g)_+/p$ can be problematic. If there is some freedom in the design (i.e. in the choice of g), we should try to minimize p. This simple acceptance-complement method has been used for generating gamma and t variates (see Ahrens and Dieter (1981,1983) and Stadlober (1981) respectively). One of the main technical obstacles encountered (and overcome) by these authors was the determination of the set on which $f(x) > g(x)$. If we have two densities that are very close, we must first verify where they cross. Often this leads to complicated equations whose solutions can only be determined numerically. These problems can be sidestepped by exploiting the added flexibility of the general acceptance-complement method.

5.3. Acceleration by avoiding the ratio computation.

The time-consuming ratio evaluation $\frac{f_1}{g}$ in the acceptance condition can be avoided some of the time if we know two easy-to-compute functions h and $h*$ with the property that

$$h(x) \leq \frac{f_1(x)}{g(x)} \leq h*(x) .$$

The IF step in the acceptance-complement algorithm can be replaced in those cases by

Squeeze step in acceptance-complement method

IF $U > h(X)$
  THEN IF $U \geq h*(X)$
    THEN Generate a random variate X with density $\frac{f_2}{p}$ where $p = \int f_2$.
    ELSE IF $U > \frac{f_1(X)}{g(X)}$
      THEN Generate a random variate X with density $\frac{f_2}{p}$ where $p = \int f_2$.
RETURN X

A similar but more spectacular acceleration is possible for the Ahrens-Dieter algorithm if one can quickly determine whether a point belongs to A, where A is a subset of $f > g$. In particular, one will find that the set on which $f > g$ often is an interval, in which case this acceleration is easy to apply.

Accelerated version of the Ahrens-Dieter algorithm

Generate a random variate X with density g.
IF $X \notin A$
  THEN
    Generate a uniform [0,1] random variate U.
    IF $U > \frac{f(X)}{g(X)}$
      THEN Generate a random variate X with density $(f-g)_+/p$.
RETURN X

With probability $P(X \in A)$, no uniform random variate is generated. Thus, what one should try to do is to choose g such that $P(X \in A)$ is maximal. This in turn

suggests choosing g such that

$$\int_{f \geq g} g$$

is large.

5.4. An example: nearly flat densities on [0,1].

We will develop a universal generator for all densities f on $[-1,1]$ which satisfy the following property: $\sup_x f(x) - \inf_x f(x) \leq \frac{1}{2}$. Because we always have $0 \leq \inf_x f(x) \leq \frac{1}{2} \leq \sup_x f(x)$, we see that $\sup_x f(x) \leq 1$. We will apply the acceptance-complement method here with as simple a decomposition as possible, for example

$$g(x) = \frac{1}{2} \qquad (|x| \leq 1);$$

$$f_1(x) = f(x) - (f_{\max} - \frac{1}{2}) \qquad (f_{\max} = \sup_x f(x));$$

$$f_2(x) = f_{\max} - \frac{1}{2} \qquad (|x| \leq 1).$$

The condition imposed on the class of densities follows from the fact that we must ask that f_1 be nonnegative. The algorithm now becomes:

Acceptance-complement method for nearly flat densities

Generate a uniform [-1,1] random variate X.
Generate a uniform [0,1] random variate U.
IF $U > 2(f(X) - f_{\max} + \frac{1}{2})$
    THEN Generate a uniform [-1,1] random variate X.
RETURN X

To this, we could add a squeeze step, because we can exit whenever $U \leq 2(\inf_x f(x) - f_{\max} + \frac{1}{2})$, and the probability of this fast exit increases with the "flatness" of f. It is 1 when f is the uniform density.

A comparison with the rejection method is in order here. First we observe that because we picked g and f_2 both uniform, we need only uniform random variates. The number N of such uniform random variates used up in the algorithm is either 2 or 3. We have

$$E(N) = 2+1\times P(U > 2(f(X)-f_{\max}+\frac{1}{2}),$$

where X stands for a uniform [−1,1] random variate. Thus,

$$E(N) = 2+\int_{-1}^{1}\frac{1}{2}2(f_{\max}-f(x))\,dx$$

$$= 2+2f_{\max}-1 = 1+2f_{\max}.$$

In addition, if no squeeze step is used, we require exactly one computation of f per variate. The obvious rejection algorithm for this example is

Rejection algorithm for nearly flat densities

REPEAT
  Generate a uniform [−1,1] random variate X.
  Generate a uniform [0,1] random variate U.
UNTIL $Uf_{\max} \le f(X)$
RETURN X

Here too we could insert a squeeze step ($Uf_{\max} \le \inf_x f(x)$). Without it, the expected number of uniform random variates needed is 2 times the expected number of iterations, i.e. $4f_{\max}$. In addition, the expected number of computations of f is $2f_{\max}$. On both counts, this is strictly worse than the acceptance-complement method.

We have thus established that for some fairly general classes of densities, we have a strict improvement over the rejection algorithm. The universality of the algorithms depends upon the knowledge of the infimum and supremum of f. This is satisfied for example if we know that f is symmetric unimodal in which case the infimum is $f(1)$ and the supremum is $f(0)$.

The algorithm given above can be applied to the main body of most symmetric unimodal densities such as the normal and Cauchy densities. For the truncated Cauchy density

$$f(x) = \frac{2}{\pi(1+x^2)} \qquad (|x| \le 1),$$

our conditions are satisfied because $f_{\max} = \frac{2}{\pi}$ and the infimum of f is $\frac{1}{\pi}$, the difference being smaller than $\frac{1}{2}$. In this case, the expected number of uniform random variates needed is $1+\frac{4}{\pi}$. Next, note that if we can generate a random variate X with density f, then a standard Cauchy random variate can be obtained by exploiting the property that the random variate Y defined by

$$Y = \begin{cases} X & \text{with probability } \frac{1}{2} \\ \frac{1}{X} & \text{with probability } \frac{1}{2} \end{cases}$$

is Cauchy distributed. For this, we need an extra coin flip. Usually, extra coin flips are generated by borrowing a random bit from U. For example, in the universal algorithm shown above, we could have started from a uniform $[-1,1]$ random variate U, and used $|U|$ in the acceptance condition. Since $\text{sign}(U)$ is independent of $|U|$, $\text{sign}(U)$ can be used to replace X by $\frac{1}{X}$, so that the returned random variate has the standard Cauchy density. The Cauchy generator thus obtained was first developed by Kronmal and Peterson (1981).

We were forced by technical considerations to limit the densities somewhat. The rejection method can be used on all bounded densities with compact support. This typifies the situation in general. In the acceptance-complement method, once we choose the general form of g and f_2, we loose in terms of universality. For example, if both f_2 and g are constant on $[-1,1]$, then $f = f_1 + f_2 \leq g + f_2 \leq 1$. Thus, no density f with a peak higher than 1 can be treated by the method. If universality is a prime concern, then the rejection method has little competition.

5.5. Exercises.

1. Kronmal and Peterson (1981) developed yet another Cauchy generator based upon the acceptance-complement method. It is based upon the following decomposition of the truncated Cauchy density f (see text for the definition) into $f_1 + f_2$:

$$f_1(x) = f(x) - \frac{1}{\pi}(1 - |x|) \quad (|x| \leq 1);$$

$$f_2(x) = \frac{1}{\pi}(1 - |x|) \quad (|x| \leq 1);$$

$$g(x) = \frac{1}{2} \quad (|x| \leq 1).$$

We have:

A Cauchy generator of Kronmal and Peterson

Generate iid uniform [−1,1] random variates X and U.

IF $|U| > \frac{2}{\pi}$.

THEN IF $|U| \leq 0.7225$

THEN IF $|U| > \frac{4}{\pi(1+X^2)} - \frac{2}{\pi}(1-|X|)$

THEN Generate iid uniform [−1,1] random variates X,U.
$X \leftarrow |X| - |U|$.

ELSE Generate a uniform [−1,1] random variate U.
$X \leftarrow |X| - |U|$.

IF $U \leq 0$

THEN RETURN X

ELSE RETURN $\frac{1}{X}$

The first two IF's are not required for the algorithm to be correct: they correspond to squeeze steps. Verify that the algorithm generates standard Cauchy random variates. Prove also that the acceleration steps are valid. The constant 0.7225 is but an approximation of an irrational number, which should be determined.