with obvious changes if the second subinterval $[(a_{i-1}-a_{i+1})/(b_{i+1}-b_{i-1}), x_{i+1}]$ is chosen.

The right panel of Figure 2.8 shows the good agreement between the histogram and density of g_n. (See Problems 2.37 and 2.38 for generating the g_n corresponding to $\pi(b|\mathbf{x}, \mathbf{y}, a)$, and Problems 9.7 and 9.8 for full Gibbs samplers.) ||

2.5 Problems

2.1 Check the uniform random number generator on your computer:
(a) Generate 1,000 uniform random variables and make a histogram
(b) Generate uniform random variables $(X_1, \ldots, X_n)$ and plot the pairs (X_i, X_{i+1}) to check for autocorrelation.

2.2 (a) Generate a binomial $\mathcal{B}in(n, p)$ random variable with $n = 25$ and $p = .2$. Make a histogram and compare it to the binomial mass function, and to the R binomial generator.
(b) Generate 5,000 *logarithmic series* random variables with mass function

$$P(X = x) = \frac{-(1-p)^x}{x \log p}, \quad x = 1, 2, \ldots \quad 0 < p < 1.$$

Make a histogram and plot the mass function.

2.3 In each case generate the random variables and compare to the density function
(a) Normal random variables using a Cauchy candidate in Accept–Reject;
(b) Gamma $\mathcal{G}a(4.3, 6.2)$ random variables using a Gamma Ga(4, 7);
(c) Truncated normal: Standard normal truncated to $(2, \infty)$.

2.4 The arcsine distribution was discussed in Example 2.2.
(a) Show that the *arcsine distribution*, with density $f(x) = 1/\pi\sqrt{x(1-x)}$, is invariant under the transform $y = 1 - x$, that is, $f(x) = f(y)$.
(b) Show that the uniform distribution $\mathcal{U}_{[0,1]}$ is invariant under the "tent" transform,

$$D(x) = \begin{cases} 2x & \text{if } x \le 1/2 \\ 2(1-x) & \text{if } x > 1/2. \end{cases}$$

(c) As in Example 2.2, use both the arcsine and "tent" distributions in the dynamic system $X_{n+1} = D(X_n)$ to generate 100 uniform random variables. Check the properties with marginal histograms, and plots of the successive iterates.
(d) The tent distribution can have disastrous behavior. Given the finite representation of real numbers in the computer, show that the sequence (X_n) will converge to a fixed value, as the tent function progressively eliminates the last decimals of X_n. (For example, examine what happens when the sequence starts at a value of the form $1/2^n$.)

2.5 For each of the following distributions, calculate the explicit form of the distribution function and show how to implement its generation starting from a uniform random variable: (*a*) exponential; (*b*) double exponential; (*c*) Weibull; (*d*) Pareto; (*e*) Cauchy; (*f*) extreme value; (*g*) arcsine.

2.6 Referring to Example 2.7:
(a) Show that if $U \sim \mathcal{U}_{[0,1]}$, then $X = -\log U/\lambda \sim \mathcal{E}xp(\lambda)$.
(b) Verify the distributions in (2.2).
(c) Show how to generate an $\mathcal{F}_{m,n}$ random variable, where both m and n are even integers.
(d) Show that if $U \sim \mathcal{U}_{[0,1]}$, then $X = \log \frac{u}{1-u}$ is a Logistic$(0,1)$ random variable. Show also how to generate a Logistic(μ, β) random variable.

2.7 Establish the properties of the *Box–Muller algorithm* of Example 2.8. If U_1 and U_2 are iid $\mathcal{U}_{[0,1]}$, show that:
(a) The transforms

$$X_1 = \sqrt{-2\log(U_1)}\,\cos(2\pi U_2)\,, \qquad X_2 = \sqrt{-2\log(U_1)}\,\sin(2\pi U_2)\,,$$

are iid $\mathcal{N}(0,1)$.
(b) The polar coordinates are distributed as

$$r^2 = X_1^2 + X_2^2 \sim \chi_2^2\,,$$
$$\theta = \arctan\frac{X_2}{X_1} \sim \mathcal{U}[0, 2\pi].$$

(c) Establish that $\exp(-r^2/2) \sim \mathcal{U}[0,1]$, and so r^2 and θ can be simulated directly.

2.8 (Continuation of Problem 2.7)
(a) Show that an alternate version of the Box–Muller algorithm is
Algorithm A.8 –Box–Muller (2)–

1. Generate [A.8]

$$U_1, U_2 \sim \mathcal{U}([-1,1])$$

   until $S = U_1^2 + U_2^2 \le 1$.
2. Define $Z = \sqrt{-2\log(S)/S}$ and take

$$X_1 = Z\,U_1, \qquad X_2 = Z\,U_2.$$

(*Hint:* Show that (U_1, U_2) is uniform on the unit sphere and that X_1 and X_2 are independent.)
(b) Give the average number of generations in `1.` and compare with the original Box–Muller algorithm [A.3] on a small experiment.
(c) Examine the effect of not constraining (U_1, U_2) to the unit circle.

2.9 Show that the following version of the Box–Muller algorithm produces one normal variable and compare the execution time with both versions [A.3] and [A.8]:

Algorithm A.9 –Box–Muller (3)–

1. Generate

$$Y_1, Y_2 \sim \mathcal{E}xp(1)$$

   until $Y_2 > (1 - Y_1)^2/2$. [A.9]
2. Generate $U \sim \mathcal{U}([0,1])$ and take

$$X = \begin{cases} Y_1 & \text{if } U < 0.5 \\ -Y_1 & \text{if } U > 0.5. \end{cases}$$

2.10 Examine the properties of an algorithm to simulate $\mathcal{N}(0,1)$ random variables based on the Central Limit Theorem, which takes the appropriately adjusted mean of a sequence of uniforms $U_1, \ldots, U_n$ for $n = 12$, $n = 48$ and $n = 96$. Consider, in particular, the moments, ranges, and tail probability calculations based on the generated variables.

2.11 For the generation of a Cauchy random variable, compare the inversion method with one based on the generation of the normal pair of the polar Box–Muller method (Problem 2.8).

(a) Show that, if X_1 and X_2 are iid normal, $Y = X_1/X_2$ is distributed as a Cauchy random variable.

(b) Show that the Cauchy distribution function is $F(x) = \tan^{-1}(x)/\pi$, so the inversion method is easily implemented.

(c) Is one of the two algorithms superior?

2.12 Use the algorithm of Example 2.10 to generate the following random variables. In each case make a histogram and compare it to the mass function, and to the generator in your computer.

(a) Binomials and Poisson distributions;

(b) The hypergeometric distribution;

(c) The *logarithmic series* distribution. A random variable X has a logarithmic series distribution with parameter p if

$$P(X = x) = \frac{-(1-p)^x}{x \log p}, \quad x = 1, 2, \ldots, \quad 0 < p < 1.$$

(d) Referring to part (a), for different parameter values, compare the algorithms there with those of Problems 2.13 and 2.16.

2.13 Referring to Example 2.9.

(a) Show that if $N \sim \mathcal{P}(\lambda)$ and $X_i \sim \mathcal{E}xp(\lambda)$, $i \in \mathbb{N}^*$, independent, then

$$P_\lambda(N = k) = P_\lambda(X_1 + \cdots + X_k \le 1 < X_1 + \cdots + X_{k+1}) .$$

(b) Use the results of part (a) to justify that the following algorithm simulates a Poisson $\mathcal{P}(\lambda)$ random variable:

Algorithm A.10 –Poisson simulation–

```
   p = 1,  N = 0,  c = e^{-λ}.
1. Repeat                                    [A.10]
      N = N + 1
      generate U_i
      update p = pU_i
   until p < c.
2. Take X = N - 1 .
```

(*Hint:* For part (a), integrate the Gamma density by parts.)

2.14 There are (at least) two ways to establish (2.4) of Example 2.12:

(a) Make the transformation $x = yu^{1/\alpha}$, $w = y$, and integrate out w.

(b) Make the transformation $x = yz$, $w = z$, and integrate out w.

2.15 In connection with Example 2.16, for a Beta distribution $\mathcal{B}e(\alpha, \beta)$, find the maximum of the $\mathcal{B}e(\alpha, \beta)$ density.

2.16 Establish the validity of Knuth (1981) $\mathcal{B}(n, p)$ generator:

Algorithm A.11 –Binomial–

Define $k = n$, $\theta = p$ and $x = 0$.
1. Repeat $\quad i = [1 + k\theta]$
   $V \sim \mathcal{B}e(i, k + 1 - i)$ $\quad [A.11]$
   if $\theta > V$, $\theta = \theta/V$ and $k = i - 1$;
   otherwise, $x = x + i$, $\theta = (\theta - V)/(1 - V)$ and $k = k - i$
   until $k \leq K$.
2. For $i = 1, 2, \ldots, k$,
   generate U_i
   if $U_i < p$, $x = x + 1$.
3. Take x.

2.17 Establish the claims of Example 2.11: If $U_1, \ldots, U_n$ is an iid sample from $\mathcal{U}_{[0,1]}$ and $U_{(1)} \leq \cdots \leq U_{(n)}$ are the corresponding order statistics, show that
(a) $U_{(i)} \sim \mathcal{B}e(i, n - i + 1)$;
(b) $(U_{(i_1)}, U_{(i_2)} - U_{(i_1)}, \ldots, U_{(i_k)} - U_{(i_{k-1})}, 1 - U_{(i_k)}) \sim \mathcal{D}(i_1, i_2 - i_1, \ldots, n - i_k + 1)$;
(c) If U and V are iid $\mathcal{U}_{[0,1]}$, the distribution of

$$\frac{U^{1/\alpha}}{U^{1/\alpha} + V^{1/\beta}},$$

conditional on $U^{1/\alpha} + V^{1/\beta} \leq 1$, is the $\mathcal{B}e(\alpha, \beta)$ distribution.
(d) Show that the order statistics can be directly generated via the *Renyi representation* $u_{(i)} = \sum_{j=1}^{i} \nu_j / \sum_{j=1}^{n} \nu_j$, where the ν_j's are iid $\mathcal{E}xp(1)$.

2.18 For the generation of a Student's t distribution, $\mathcal{T}(\nu, 0, 1)$, Kinderman et al. (1977) provide an alternative to the generation of a normal random variable and a chi squared random variable.

Algorithm A.12 –Student's t–

1. `Generate` $U_1, U_2 \sim \mathcal{U}([0,1])$.
2. `If` $U_1 < 0.5$, $X = 1/(4U_1 - 1)$ `and` $V = X^{-2}U_2$;
   `otherwise,` $X = 4U_1 - 3$ `and` $V = U_2$. $\quad [A.12]$
3. `If` $V < 1 - (|X|/2)$ `or` $V < (1 + (X^2/\nu))^{-(\nu+1)/2}$, `take` X;
   `otherwise, repeat.`

Validate this algorithm and compare it with the algorithm of Example 2.13.

2.19 For $\alpha \in [0,1]$, show that the algorithm

Algorithm A.13

`Generate`
$U \sim \mathcal{U}([0,1])$ $\quad [A.13]$
`until` $U < \alpha$.

produces a simulation from $\mathcal{U}([0, \alpha])$. Compare it with the transform αU, $U \sim \mathcal{U}(0,1)$ for values of α close to 0 and close to 1.

2.20 In each case generate the random variables and compare the histogram to the density function
(a) Normal random variables using a Cauchy candidate in Accept–Reject
(b) Gamma$(4.3, 6.2)$ random variables using a Gamma$(4, 7)$.

(*c*) Truncated normal: Standard normal truncated to $(2, \infty)$

2.21 An efficient algorithm for the simulation of Gamma $\mathcal{G}a(\alpha, 1)$ distributions is based on *Burr's distribution*, a distribution with density

$$g(x) = \lambda\mu \, \frac{x^{\lambda-1}}{(\mu + x^{\lambda})^2} \, , \qquad x > 0 \, .$$

It has been developed by Cheng (1977) and Cheng and Feast (1979). (See Devroye 1985.) For $\alpha > 1$, it is

Algorithm A.14 –Cheng and Feast's Gamma–

```
   Define c_1 = α−1, c_2 = (α−(1/6α))/c_1, c_3 = 2/c_1, c_4 = 1+c_3,
   and c_5 = 1/√α.
1. Repeat
   generate U_1, U_2
   take U_1 = U_2 + c_5(1 − 1.86U_1) if α > 2.5
   until 0 < U_1 < 1.                                          [A.14]
2. W = c_2 U_2/U_1.
3. If c_3 U_1 + W + W^{-1} ≤ c_4 or c_3 log U_1 − log W + W ≤ 1,
   take c_1 W;
   otherwise, repeat.
```

(a) Show g is a density.
(b) Show that this algorithm produces variables generated from $\mathcal{G}a(\alpha, 1)$.

2.22 Ahrens and Dieter (1974) propose the following algorithm to generate a Gamma $\mathcal{G}a(\alpha, 1)$ distribution:

Algorithm A.15 –Ahrens and Dieter's Gamma–

```
1. Generate U_0, U_1.
2. If U_0 > e/(e+α), x = −log{(α+e)(1−U_0)/αe} and y = x^{α−1};
   otherwise, x = {(α+e)U_0/e}^{1/α} and y = e^{−x}.          [A.15]
3. If U_1 < y, take x;
   otherwise, repeat.
```

Show that this algorithm produces variables generated from $\mathcal{G}a(\alpha, 1)$. Compare with Problem 2.21.

2.23 To generate the Beta distribution $\mathcal{B}e(\alpha, \beta)$ we can use the following representation:

(a) Show that, if $Y_1 \sim \mathcal{G}a(\alpha, 1)$, $Y_2 \sim \mathcal{G}a(\beta, 1)$, then

$$X = \frac{Y_1}{Y_1 + Y_2} \sim \mathcal{B}e(\alpha, \beta) \, .$$

(b) Use part (a) to construct an algorithm to generate a Beta random variable.
(c) Compare this algorithm with the method given in Problem 2.17 for different values of (α, β).
(d) Compare this algorithm with an Accept–Reject algorithm based on (i) the uniform distribution; (ii) the truncated normal distribution (when $\alpha \geq 1$ and $\beta \geq 1$).

(*Note:* See Schmeiser and Shalaby 1980 for an alternative Accept–Reject algorithm to generate Beta rv's.)

2.24 (a) Show that Student's t density can be written in the form (2.6), where $h_1(x|y)$ is the density of $\mathcal{N}(0, \nu/y)$ and $h_2(y)$ is the density of χ^2_ν.
(b) Show that Fisher's $\mathcal{F}_{m,\nu}$ density can be written in the form (2.6), with $h_1(x|y)$ the density of $\mathcal{G}a(m/2, \nu/m)$ and $h_2(y)$ the density of χ^2_ν.

2.25 The noncentral chi squared distribution, $\chi^2_p(\lambda)$, can be defined by a mixture representation (2.6), where $h_1(x|K)$ is the density of χ^2_{p+2K} and $h_2(k)$ is the density of $\mathcal{P}(\lambda/2)$.
(a) Show that it can also be expressed as the sum of a χ^2_{p-1} random variable and of the square of a standard normal variable.
(b) Compare the two algorithms which can be derived from these representations.
(c) Discuss whether a direct approach via an Accept–Reject algorithm is at all feasible.

2.26 (Walker 1997) Show that the Weibull distribution, $\mathcal{W}e(\alpha, \beta)$, with density

$$f(x|\alpha, \beta) = \beta\alpha x^{\alpha-1} \exp\left(-\beta x^\alpha\right),$$

can be represented as a mixture of $X \sim \mathcal{B}e(\alpha, \omega^{1/\alpha})$ by $\omega \sim \mathcal{G}a(2, \beta)$. Examine whether this representation is helpful from a simulation point of view.

2.27 An application of the mixture representation can be used to establish the following result (see Note 2.6.3):

Lemma 2.27. *If*

$$f(x) = \frac{f_1(x) - \varepsilon f_2(x)}{1-\varepsilon},$$

where f_1 and f_2 are probability densities such that $f_1(x) \geq \varepsilon f_2(x)$, the algorithm

Generate

$$(X, U) \sim f_1(x)\, \mathbb{I}_{[0,1]}(u)$$

until $U \geq \varepsilon f_2(X)/f_1(X)$.

produces a variable X distributed according to f.

(a) Show that the distribution of X satisfies

$$P(X \leq x_0) = \int_{-\infty}^{x_0} \left(1 - \frac{\varepsilon f_2(x)}{f_1(x)}\right) f_1(x)\, dx \sum_{i=0}^{\infty} \varepsilon^i .$$

(b) Evaluate the integral in (a) to complete the proof.

2.28 (a) Demonstrate the equivalence of Corollary 2.17 and the Accept–Reject algorithm.
(b) Generalize Corollary 2.17 to the multivariate case. That is, for $\mathbf{X} = (X_1, X_2, \ldots, X_p) \sim f(x_1, x_2, \ldots, x_p) = f(\mathbf{x})$, formulate a joint distribution $(\mathbf{X}, U) \sim \mathbb{I}(0 < u < f(\mathbf{x}))$, and show how to generate a sample from $f(\mathbf{x})$ based on uniform random variables.

2.29 (a) Referring to (2.10), if $\pi(\theta|x)$ is the target density in an Accept–Reject algorithm, and $\pi(\theta)$ is the candidate density, show that the bound M can be taken to be the likelihood function evaluated at the MLE.

(b) For estimating a normal mean, a robust prior is the Cauchy. For $X \sim N(\theta, 1)$, $\theta \sim \text{Cauchy}(0, 1)$, the posterior distribution is

$$\pi(\theta|x) \propto \frac{1}{\pi(1+\theta^2)} \frac{1}{2\pi} e^{-(x-\theta)^2/2}.$$

Use the Accept–Reject algorithm, with a Cauchy candidate, to generate a sample from the posterior distribution.

(*Note:* See Problem 3.19 and Smith and Gelfand 1992.)

2.30 For the Accept–Reject algorithm [A.4], with f and g properly normalized,

(a) Show that the probability of accepting a random variable is

$$P\left(U < \frac{f(X)}{Mg(X)}\right) = \frac{1}{M}.$$

(b) Show that $M \geq 1$.

(c) Let N be the number of trials until the kth random variable is accepted. Show that, for the normalized densities, N has the negative binomial distribution $\mathcal{N}eg(k, p)$, where $p = 1/M$. Deduce that the expected number of trials until k random variables are obtained is kM.

(d) Show that the bound M does not have to be tight; that is, there may be $M' < M$ such that $f(x) \leq M'g(x)$. Give an example where it makes sense to use M instead of M'.

(e) When the bound M is too tight (i.e., when $f(x) > Mg(x)$ on a non-negligible part of the support of f), show that the algorithm $[A.4]$ does not produce a generation from f. Give the resulting distribution.

(f) When the bound is not tight, show that there is a way, using Lemma 2.27, to recycle part of the rejected random variables. (*Note:* See Casella and Robert 1998 for details.)

2.31 For the Accept–Reject algorithm of the $\mathcal{G}a(n, 1)$ distribution, based on the $\mathcal{E}xp(\lambda)$ distribution, determine the optimal value of λ.

2.32 This problem looks at a generalization of Example 2.19.

(a) If the target distribution of an Accept–Reject algorithm is the Gamma distribution $\mathcal{G}a(\alpha, \beta)$, where $\alpha \geq 1$ is not necessarily an integer, show that the instrumental distribution $\mathcal{G}a(a, b)$ is associated with the ratio

$$\frac{f(x)}{g(x)} = \frac{\Gamma(a)}{\Gamma(\alpha)} \frac{\beta^\alpha}{b^a} x^{\alpha-a} e^{-(\beta-b)x}.$$

(b) Why do we need $a < \alpha$ and $b < \beta$?

(c) For $a = \lfloor \alpha \rfloor$, show that the bound is maximized (in x) at $x = (\alpha - a)/(\beta - b)$.

(d) For $a = \lfloor \alpha \rfloor$, find the optimal choice of b.

(e) Compare with $a' = \lfloor \alpha \rfloor - 1$, when $\alpha > 2$.

2.33 The right-truncated Gamma distribution $\mathcal{TG}(a, b, t)$ is defined as the restriction of the Gamma distribution $\mathcal{G}a(a, b)$ to the interval $(0, t)$.

(a) Show that we can consider $t = 1$ without loss of generality.

(b) Give the density f of $\mathcal{TG}(a, b, 1)$ and show that it can be expressed as the following mixture of Beta $\mathcal{B}e(a, k+1)$ densities:

$$f(x) = \sum_{k=0}^{\infty} \frac{b^a e^{-b}}{\gamma(a, b)} \frac{b^k}{k!} x^{a-1} (1-x)^k,$$

where $\gamma(a, b) = \int_0^1 x^{a-1} e^{-x}$.

(c) If f is replaced with g_n which is the series truncated at term $k=n$, show that the acceptance probability of the Accept–Reject algorithm based on (g_n, f) is

$$\frac{1-\dfrac{\gamma(n+1,b)}{n!}}{1-\dfrac{\gamma(a+n+1,b)\Gamma(a)}{\Gamma(a+n+1)\gamma(a,b)}}\,.$$

(d) Evaluate this probability for different values of (a,b).
(e) Give an Accept–Reject algorithm based on the pair (g_n, f) and a computable bound. (*Note:* See Philippe 1997c for a complete resolution of the problem.)

2.34 Let $f(x)=\exp(-x^2/2)$ and $g(x)=1/(1+x^2)$, densities of the normal and Cauchy distributions, respectively (ignoring the normalization constants).

(a) Show that the ratio

$$\frac{f(x)}{g(x)}=(1+x^2)\,e^{-x^2/2}\le 2/\sqrt{e},$$

which is attained at $x=\pm 1$.
(b) Show that for the normalized densities, the probability of acceptance is $\sqrt{e/2\pi}=0.66.$, which implies that, on the average, one out of every three simulated Cauchy variables is rejected. Show that the mean number of trials to success is $1/.66=1.52$.
(c) Replacing g by a Cauchy density with scale parameter σ,

$$g_\sigma(x)=1/\{\pi\sigma(1+x^2/\sigma^2)\}\;,$$

show that the bound on f/g_σ is $2\sigma^{-1}\exp\{\sigma^2/2-1\}$ and is minimized by $\sigma^2=1$. (This shows that $\mathcal{C}(0,1)$ is the best choice among the Cauchy distributions for simulating a $\mathcal{N}(0,1)$ distribution.)

2.35 There is a direct generalization of Corollary 2.17 that allows the proposal density to change at each iteration.

Algorithm A.16 –Generalized Accept–Reject–

At iteration i $(i\ge 1)$
1. Generate $X_i\sim g_i$ and $U_i\sim\mathcal{U}([0,1])$, independently.
2. If $U_i\le \epsilon_i f(X_i)/g_i(X_i)$, accept $X_i\sim f$;
3. otherwise, move to iteration $i+1$.

(a) Let Z denote the random variable that is output by this algorithm. Show that Z has the cdf

$$P(Z\le z)=\sum_{i=1}^{\infty}\epsilon_i\prod_{j=1}^{i-1}(1-\epsilon_j)\int_{-\infty}^{z}f(x)dx\,.$$

(b) Show that

$$\sum_{i=1}^{\infty}\epsilon_i\prod_{j=1}^{i-1}(1-\epsilon_j)=1 \text{ if and only if } \sum_{i=1}^{\infty}\log(1-\epsilon_i)\text{ diverges}\,.$$

Deduce that we have a valid algorithm if the second condition is satisfied.

(c) Give examples of sequences ϵ_i that satisfy, and do not satisfy, the requirement of part (b).

2.36 (a) Prove the validity of the ARS Algorithm [A.7], without the envelope step, by applying Algorithm [A.16].

(b) Prove the validity of the ARS Algorithm [A.7], with the envelope step, directly. Note that

$$P(X \le x|\text{Accept}) = P\left(X \le x \middle| \left\{U < \frac{g_\ell}{Mg_m} \text{ or } U < \frac{f}{Mg_m}\right\}\right)$$

and

$$\left\{U < \frac{g_\ell}{Mg_m} \text{ or } U < \frac{f}{Mg_m}\right\} = \left\{U < \frac{g_\ell}{Mg_m}\right\} \cup \left\{\frac{g_\ell}{Mg_m} < U < \frac{f}{Mg_m}\right\},$$

which are disjoint.

2.37 Based on the discussion in Example 2.26, write an alternate algorithm to Algorithm [A.17] that does not require the calculation of the density g_n.

2.38 The histogram and density of Figure 2.8 give the candidate g_n for $\pi(a|\mathbf{x}, \mathbf{y}, b)$, the conditional density of the intercept a in $\log \lambda = a + bt$, where we set $b = .025$ and $\sigma^2 = 5$. Produce the same picture for the slope, b, when we set the intercept $a = .15$ and $\tau^2 = 5$.

2.39 Step `1` of [A.7] relies on simulations from g_n. Show that we can write

$$g_n = \varpi_n^{-1} \left\{ \sum_{i=0}^{r_n} e^{\alpha_i x + \beta_i} \, \mathbb{I}_{[x_i, x_{i+1}]}(x) + e^{\alpha_{-1} x + \beta_{-1}} \mathbb{I}_{[-\infty, x_0]}(x) \right.$$
$$\left. e^{\alpha_{r_n+1} x + \beta_{r_n+1}} \, \mathbb{I}_{[x_{n+1}, +\infty]}(x) \right\},$$

where $y = \alpha_i x + \beta_i$ is the equation of the segment of line corresponding to g_n on $[x_i, x_{i+1}]$, r_n denotes the number of segments, and

$$\begin{aligned} \varpi_n &= \int_{-\infty}^{x_0} e^{\alpha_{-1} x + \beta_{-1}} dx + \sum_{i=0}^{n} \int_{x_i}^{x_{i+1}} e^{\alpha_i x + \beta_i} dx + \int_{x_{n+1}}^{+\infty} e^{\alpha_{r_n+1} x + \beta_{r_n+1}} dx \\ &= \frac{e^{\alpha_{-1} x_0 + \beta_{-1}}}{\alpha_{-1}} + \sum_{i=0}^{n} e^{\beta_i} \frac{e^{\alpha_i x_{i+1}} - e^{\alpha_i x_i}}{\alpha_i} - \frac{e^{\alpha_{r_n+1} x_{n+1}}}{\alpha_{r_n+1}}, \end{aligned}$$

when $\operatorname{supp} f = \mathbb{R}$.

Verify that this representation as a sequence validates the following algorithm for simulation from g_n:

Algorithm A.17 –Supplemental ARS Algorithm–

1. Select the interval $[x_i, x_{i+1}]$ with probability

$$e^{\beta_i} \frac{e^{\alpha_i x_{i+1}} - e^{\alpha_i x_i}}{\varpi_n \, \alpha_i}. \qquad [A.17]$$

2. Generate $U \sim \mathcal{U}_{[0,1]}$ and take

$$X = \alpha_i^{-1} \log[e^{\alpha_i x_i} + U(e^{\alpha_i x_{i+1}} - e^{\alpha_i x_i})].$$

Note that the segment $\alpha_i + \beta_i x$ is not the same as the line $a_i + b_i x$ used in (2.16)

2.40 As mentioned in Section 2.4, many densities are log-concave.

(a) Show that the so-called *natural* exponential family,

$$dP_\theta(x) = \exp\{x \cdot \theta - \psi(\theta)\} d\nu(x)$$

is log-concave.

(b) Show that the logistic distribution of (2.23) is log-concave.

(c) Show that the *Gumbel* distribution

$$f(x) = \frac{k^k}{(k-1)!} \exp\left\{-kx - ke^{-x}\right\} , \quad k \in \mathbb{N}^* ,$$

is log-concave (Gumbel 1958).

(d) Show that the *generalized inverse Gaussian* distribution,

$$f(x) \propto x^\alpha \, e^{-\beta x - \alpha/x} , \qquad x > 0, \ \alpha > 0, \ \beta > 0 ,$$

is log-concave.

2.41 (George et al. 1993) For the natural exponential family, the conjugate prior measure is defined as

$$d\pi(\theta|x_0, n_0) \propto \exp\{x_0 \cdot \theta - n_0 \psi(\theta)\} d\theta,$$

with $n_0 > 0$. (See Brown 1986, Chapter 1, for properties of exponential families.)

(a) Show that

$$\varphi(x_0, n_0) = \log \int_\Theta \exp\{x_0 \cdot \theta - n_0 \psi(\theta)\} d\theta$$

is convex.

(b) Show that the so-called *conjugate likelihood distribution*

$$L(x_0, n_0|\theta_1, \ldots, \theta_p) \propto \exp\left\{x_0 \cdot \sum_{i=1}^p \theta_i - n_0 \sum_{i=1}^p \psi(\theta) - p\varphi(x_0, n_0)\right\}$$

is log-concave in (x_0, n_0).

(c) Deduce that the ARS algorithm applies in hierarchical Bayesian models with conjugate priors on the natural parameters and log-concave hyperpriors on (x_0, n_0).

(d) Apply the ARS algorithm to the case

$$X_i|\theta_i \sim \mathcal{P}(\theta_i t_i), \qquad \theta_i \sim \mathcal{G}a(\alpha, \beta), \qquad i = 1, \ldots, n,$$

with fixed α and $\beta \sim \mathcal{G}a(0.1, 1)$.

2.42 In connection with Example 2.25,

(a) Show that a sum of log-concave functions is a log-concave function.

(b) Deduce that (2.15) is log-concave.

2.43 (Casella and Berger 2001, Section 8.3) This problem examines the relationship of the property of log-concavity with other desirable properties of density functions.

(a) The property of *monotone likelihood ratio* is very important in the construction of hypothesis tests, and in many other theoretical investigations. A family of pdfs or pmfs $\{g(t|\theta) \colon \theta \in \Theta\}$ for a univariate random variable T with real-valued parameter θ has a *monotone likelihood ratio* (MLR) if, for every $\theta_2 > \theta_1$, $g(t|\theta_2)/g(t|\theta_1)$ is a monotone (nonincreasing or nondecreasing) function of t on $\{t \colon g(t|\theta_1) > 0 \text{ or } g(t|\theta_2) > 0\}$. Note that $c/0$ is defined as ∞ if $0 < c$.

Show that if a density is log-concave, it has a monotone likelihood ratio.

(b) Let $f(x)$ be a pdf and let a be a number such that, if $a \geq x \geq y$ then $f(a) \geq f(x) \geq f(y)$ and, if $a \leq x \leq y$ then $f(a) \geq f(x) \geq f(y)$. Such a pdf is called *unimodal* with a *mode* equal to a.
Show that if a density is log-concave, it is unimodal.

2.44 This problem will look into one of the failings of congruential generators, the production of parallel lines of output. Consider a congruential generator $D(x) = ax \bmod 1$, that is, the output is the fractional part of ax.

(a) For $k = 1, 2, \ldots, 333$, plot the pairs $(k*0.003, D(k*0.003))$ for $a = 5, 20, 50$. What can you conclude about the parallel lines?

(b) Show that each line has slope a and the lines repeat at intervals of $1/a$ (hence, larger values of a will increase the number of lines). (*Hint:* Let $x = \frac{i}{a} + \delta$, for $i = 1, \ldots, a$ and $0 < \delta < \frac{1}{a}$. For this x, show that $D(x) = a\delta$, regardless of the value of i.)

2.6 Notes

2.6.1 The Kiss Generator

Although this book is not formally concerned with the generation of uniform random variables (as we start from the assumption that we have an endless supply of such variables), it is good to understand the basic workings and algorithms that are used to generate these variables. In this note we describe the way in which uniform pseudo-random numbers are generated, and give a particularly good algorithm.

To keep our presentation simple, rather than give a catalog of random number generators, we only give details for a single generator, the Kiss algorithm of Marsaglia and Zaman (1993). For details on other random number generators, the books of Knuth (1981), Rubinstein (1981), Ripley (1987), and Fishman (1996) are excellent sources.

As we have remarked before, the finite representation of real numbers in a computer can radically modify the behavior of a dynamic system. Preferred generators are those that take into account the specifics of this representation and provide a uniform sequence. It is important to note that such a sequence does not really take values in the interval $[0, 1]$ but rather on the integers $\{0, 1, \ldots, M\}$, where M is the largest integer accepted by the computer. One manner of characterizing the performance of these integer generators is through the notion of *period*.

Definition 2.28. The *period*, T_0, of a generator is the smallest integer T such that $u_{i+T} = u_i$ for every i; that is, such that D^T is equal to the identity function.

The period is a very important parameter, having direct impact on the usefulness of a random number generator. If the number of needed generations exceeds the period of a generator, there may be noncontrollable artifacts in the sequence (cyclic phenomena, false orderings, etc.). Unfortunately, a generator of the form $X_{n+1} = f(X_n)$ has a period no greater than $M+1$, for obvious reasons. In order to overcome this bound, a generator must utilize several sequences X_n^i simultaneously (which is a characteristic of Kiss) or must involve $X_{n-1}, X_{n-2}, \ldots$ in addition to X_n, or must use other methods such as *start-up tables*, that is, using an auxiliary table of random digits to restart the generator.

Kiss simultaneously uses two generation techniques, namely *congruential* generation and *shift register* generation.