Chapter 2
Sampling from Known Distributions

In this chapter, we give an overview of different methods that can be used to generate random variates from a given distribution. Even if inversion should be the preferred choice for quasi–Monte Carlo users, it is important to be aware of other methods that are available for that purpose. First of all, inversion is sometimes slower and more difficult to apply than other methods. In such cases, Monte Carlo users may prefer these other methods. Also, when working with predefined functions (e.g., `randn` in Matlab) to generate observations from a given distribution, it is quite possible that the underlying method is not based on inversion. In addition, there are applications for which the common approach used by people working in that area is to use something other than inversion (e.g., in computer graphics, for ray generation). In such cases, even if ultimately the quasi–Monte Carlo user will try to use inversion instead of these other methods in order to modify code or algorithms appropriately, it is important to understand what the other method does. Finally, in some cases inversion may not be directly applicable, and an alternative method needs to be used.

We assume the reader is familiar with common distributions such as those already encountered in Chap. 1 — exponential, gamma, binomial, and normal — and will not describe specifically how to handle each one of these in this chapter. Instead, we wish to describe general techniques that can be used for a variety of models. More precisely, we describe four general approaches that can be used for generating random variates from a given (univariate) distribution and then talk about the multivariate case. Much more extensive coverage of specific distributions and algorithms can be found in [45, 75, 196, 243, 391]. In particular, Luc Devroye's book (which is out of print) can be downloaded from his Web page [485].

Before we do this, we want to briefly discuss a few distributions that are often encountered in simulation models.

C. Lemieux, *Monte Carlo and Quasi–Monte Carlo Sampling*,

Springer Series in Statistics 692, DOI: 10.1007/978-0-387-78165-5_2,

2.1 Common distributions arising in stochastic models

Our goal in this section is simply to talk about a few distributions that are commonly used in stochastic models. Our discussion is by no means extensive, as we restrict ourselves to distributions arising in the different examples used throughout the book.

Normal and Lognormal Distribution

The normal distribution arises very often in financial simulation models. We already saw an example in Sect. 1.6 when discussing equity-linked contracts. One reason why it arises so often is that the *Brownian motion* is often used as a building block to model asset prices, and the increments of a Brownian motion are normally distributed. Because of the importance of this process, we give a formal definition before going further. The reader is referred to [212, 350, 388] for more information.

Definition 2.1. A *standard Brownian motion* is a continuous-time stochastic process $\{B(t), t \geq 0\}$ with the following properties:

1. $B(0) = 0$.
2. The increments over disjoint intervals are independent. That is, for $r < s < t < u$, $B(u) - B(t)$ and $B(s) - B(r)$ are independent.
3. The increments are stationary. That is, for any $r, s, t > 0$, $B(r+t) - B(r)$ and $B(s+t) - B(s)$ have the same probability function, which is normal with mean $\mu = 0$ and variance t.

If $\{B(t), t \geq 0\}$ is a standard Brownian motion, then for $\sigma > 0$ and $\mu \in \mathbb{R}$, the process $\{\sigma B(t) + \mu t, t \geq 0\}$ is a Brownian motion with *drift* μ and *diffusion coefficient* σ.

The simplest financial model that uses a Brownian motion is the lognormal model encountered in Chap. 1, which amounts to having the asset price $S(t)$ at time t given by

$$S(t) = S(0) \exp\left((\mu - \sigma^2/2)t + \sigma B(t)\right),$$

where μ and σ are the instantaneous return rate and volatility of the asset price, respectively. Since $B(t) \sim N(0, t)$, we have that $S(t)$ has a lognormal distribution with parameters $((\mu - \sigma^2/2)t, \sigma^2 t)$.

In financial simulations, the multinormal distribution is also often encountered either when modeling a vector of financial assets — in which case they are driven by Brownian motions that are correlated — or when looking at a given asset value at different times.

Exponential, Gamma, Weibull, and Poisson distributions

The exponential distribution is frequently encountered in simulation models, partly because Poisson processes are often used to model stochastic processes that count the occurrence of a certain event — for example, client arrivals in a queue, molecular reactions in a chemical system, claims arrivals for an insurance company — and in this case the interarrival time between two events is known to have an exponential distribution.

The gamma distribution shows up in financial models that include jumps, as we discuss in Sect. 7.2 of our chapter on financial applications. It also arises as the distribution of the kth event from a Poisson process and more generally as a sum of exponential random variables. The Weibull distribution arises as the minimum of a sample of i.i.d. exponential random variables. All three distributions can also be used to model failure times.

The Poisson distribution is used to count the number of events in a Poisson process. An example was discussed in Prob. 1.17. Users may sometimes want to draw from it directly rather than generating exponential interarrival times until a certain time limit is reached. Inversion can be used to do that, and specific aspects of this task are discussed in [129].

Beta distribution

The beta distribution often arises when studying order statistics. More precisely, it comes up when we look at a sample of n i.i.d. $U(0,1)$ random variables $u_1, \ldots, u_n$, because then the ith smallest observation $u_{(i)}$ has a beta distribution with parameters $(i, n+1-i)$.

Copula-based models

Models based on copulas have become increasingly popular over the last ten years or so, for instance in biostatistics and risk management [104, 130]. Formally, a copula is a joint distribution C defined over $[0,1]^k$ and such that each marginal distribution is a $U(0,1)$. A theorem by Sklar [404] says that for any joint CDF $F(x_1, \ldots, x_k)$ with given marginal CDFs $H_1(x_1), \ldots, H_k(x_k)$, there exists a copula such that we can write

$$F(x_1, \ldots, x_k) = C(H_1(x_1), \ldots, H_k(x_k)). \tag{2.1}$$

By writing the joint CDF $F(x_1, \ldots, x_k)$ in this way, we specify the distribution in two steps. We start by choosing the marginal distributions and then introduce the dependence relation between the variables X_j via the copula function C. This formulation also naturally suggests the use of inversion to generate $(x_1, \ldots, x_k)$. We will come back to copulas in Sect. 2.6.

2.2 Inversion

This method goes back to the beginnings of Monte Carlo. It was proposed by von Neumann in a letter to Stan Ulam discussing their "random numbers work" [95]. We discussed on p. 16 of Chap. 1 how to use inversion for the exponential distribution. More generally, for a continuous distribution with CDF $F(\cdot)$, it can be applied as in Fig. 2.1.

1. $U \leftarrow$ `Rand01()`.
2. Return $X = F^{-1}(U)$.

Fig. 2.1 Steps to apply inversion for continuous distributions.

This looks very simple, but the applicability and effectiveness of this method rests on how easy it is to compute the inverse CDF F^{-1}. For the exponential, Weibull (see Prob. 2.2), and other distributions, the inverse function can be determined rather easily. But for the normal, gamma, beta, and other distributions, in particular those that do not have closed-form expressions for the corresponding CDF, inversion cannot be applied directly, and an approximation for F^{-1} must first be determined. For instance, Kennedy and Gentle discuss rational fraction approximations for the inverse CDF of a normal distribution [216, pp. 95–96]. In that setting, $F^{-1}(u)$ can be approximated by a function of the form [349]

$$F^{-1}(u) \approx t + \frac{p_0 + p_1 t + p_2 t^2 + p_3 t^3 + p_4 t^4}{q_0 + q_1 t + q_2 t^2 + q_3 t^3 + q_4 t^4}$$

for $u > 0.5$ and constants q_i, p_i, where $t = (\ln(1/u^2))^{1/2}$. The case $u < 0.5$ is handled by using the symmetry of the normal pdf, which implies that $F^{-1}(u) = -F^{-1}(1-u)$. Another well-known approximation for the inverse CDF of a normal, which is particularly popular in finance [145, p. 68], is the one proposed by Moro [324]. For other distributions, approximations have been implemented in various software packages and libraries, for example in Matlab's statistical toolbox.

For a distribution that is not continuous, inversion is applied as shown in Fig. 2.2. We give in Fig. 2.3 an example where a simple discrete distribution with $P(X = x)$ equal to 0.22, 0.16, 0.33, and 0.29 for $x = 0, 1, 2, 3$, respectively, is inverted. If U falls in the interval $[0, 0.22)$, we return $X = 0$; in $[0, 22, 0.38)$, we return $X = 1$; in $[0.38, 0.71)$, we return $X = 2$; and in $[0.71, 1]$, we return $X = 3$. This clearly causes X to have the correct distribution.

Several known discrete distributions are such that $\inf\{y : F(y) \geq u\}$ can be determined explicitly. For instance, if X has a geometric distribution with parameter p, then $P(X = x) = p(1-p)^x$, where $x \in \{0, 1, \ldots\}$. Therefore,

1. $U \leftarrow$ `Rand01()`.
2. Return $X = \inf\{y : F(y) \geq U\}$.

Fig. 2.2 Steps to apply inversion for noncontinuous distributions.

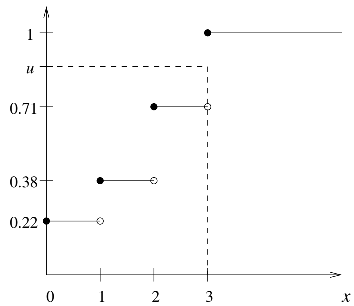

Fig. 2.3 Inverting the CDF of a discrete distribution over $\{0, 1, 2, 3\}$. The u shown is such that inversion returns $x = 3$.

$$F(x) = \sum_{y=0}^{x} p(1-p)^y = (1 - (1-p)^x),$$

and thus

$$\begin{aligned}
\inf\{y : F(y) \geq u\} &= \inf\{y : (1 - (1-p)^y) \geq u\} \\
&= \inf\{y : 1 - u \geq (1-p)^y\} \\
&= \inf\{y : (1-u)^{1/y} \geq 1 - p\} \\
&= \inf\{y : (1/y)\ln(1-u) \geq \ln(1-p)\} \\
&= \lceil \ln(1-p)/\ln(1-u) \rceil .
\end{aligned}$$

Just as in the continuous case, though, for some distributions we might not be able to derive an explicit expression for $\inf\{y : F(y) \geq u\}$. When this happens, using inversion turns out to be a searching problem, where for a given U the goal is to quickly find the index i such that

$$\sum_{j=0}^{i-1} p_j < U \leq \sum_{j=0}^{i} p_j, \tag{2.2}$$

where $p_j = P(X = x_j)$, and we assumed the domain of X was $\{x_0, x_1, \ldots\}$, where $x_j \leq x_{j+1}$ for all $j \geq 0$. (We also assumed that the sum $\sum_{j=0}^{-1} p_j = 0$.) As required, the index i satisfying (2.2) is the smallest one such that $F(x_i) \geq U$. Of course, one can perform a simple linear search starting from $i = 0$ in order to identify the correct index, but more efficient methods can (and should) be used. For instance, we can use a binary search rather than a linear one, or a "bucket scheme" meant to improve on binary search [45].

Even if inversion is sometimes slower than other methods, the fact that it uses one uniform number per random variate and transforms this number in a monotone way makes it the preferred choice when used in combination with quasi–Monte Carlo and other variance reduction techniques. As we will see below, it also works naturally well with joint distributions specified by copula functions.

2.3 Acceptance-rejection

Here the idea is to generate random variates from an alternative distribution and then accept or reject them according to a criterion designed so that overall the variates that are output have the correct distribution. More precisely, to generate random variates with a pdf $\varphi(x)$, we first find a function $t(x)$ that is majoring $\varphi(x)$ over its domain (i.e., $t(x) \geq \varphi(x)$ for all x) and whose integral is finite. Note that $t(x)$ itself usually is not a density function since

$$T = \int t(x)dx \geq \int \varphi(x)dx = 1, \tag{2.3}$$

but $r(x) := t(x)/T$ is a density function. The function $t(x)$ should be chosen so that it is easy to generate observations from $r(x)$. The algorithm described in Fig. 2.4 can then be used.

1. Generate Y having density $r(x)$.
2. Generate $U \sim U(0,1)$, independent of Y.
3. If $U \leq \varphi(Y)/t(Y)$, then return $X = Y$; otherwise go back to step 1.

Fig. 2.4 Steps for acceptance-rejection.

To understand why acceptance-rejection works, we follow the proof given in [243, App. 8A]. We first notice that each time we go through the three steps above, a pair (Y,U) is generated. To be accepted, a pair must be such that $U \leq \varphi(Y)/t(Y)$. Hence, an observation X output by this algorithm has the same distribution as $(Y|U \leq \varphi(Y)/t(Y))$; i.e., the conditional distribution of Y given that Y is accepted. Therefore,

$$P(X \leq x) = P(Y \leq x|U \leq \varphi(Y)/t(Y)) = \frac{P(Y \leq x, U \leq \varphi(Y)/t(Y))}{P(U \leq \varphi(Y)/t(Y))}.$$

Now,

$$P\left(Y \leq x, U \leq \frac{\varphi(Y)}{t(Y)}\right) = \int_{-\infty}^{x} P\left(U \leq \frac{\varphi(y)}{t(y)}\right) r(y)dy = \int_{-\infty}^{x} \frac{\varphi(y)}{t(y)} r(y)dy$$
$$= \frac{1}{T}\int_{-\infty}^{x} \varphi(y)dy = \frac{F(x)}{T},$$

where $F(x)$ is the CDF corresponding to $\varphi(x)$, and T is as defined in (2.3). In addition, we have

$$P\left(U \leq \frac{\varphi(Y)}{t(Y)}\right) = \int_{-\infty}^{\infty} \frac{\varphi(y)}{t(y)} r(y)dy = \frac{1}{T}.$$

Hence $P(X \leq x) = F(x)$, as required.

Figure 2.5 illustrates the acceptance-rejection method in the case where $\varphi(x) = 12x^2(1-x)$ for $0 \leq x \leq 1$, which corresponds to the Beta distribution with parameters $\alpha = 3$ and $\beta = 2$. Since the maximum of $\varphi(x)$ occurs at $x = 2/3$, where $\varphi(x) = 16/9$, this means we can take $t(x) = 16/9$, for $x \in [0,1]$, corresponding to a uniform density $r(x)$ over $[0,1]$. In Fig. 2.5, we show $\varphi(x)$, $t(x)$, and 200 points corresponding to trials $(Y, Ut(Y))$. When the second coordinate $Ut(Y)$ is below $\varphi(Y)$, the point is accepted; otherwise it is rejected. For this particular sample, 111 points were accepted and 89 were rejected for a proportion $111/200 = 0.555$ of acceptance, not too far from the theoretical one of $1/T = 9/16 = 0.5625$.

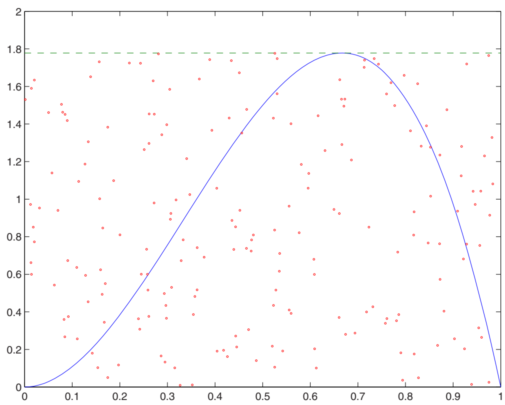

Fig. 2.5 Acceptance-rejection method for $\varphi(x) = 12x^2(1-x)$ (solid line); $t(x) = 16/9$ is the dotted line.

For practical applications, one should obviously try to use a majoring function $t(x)$ that more closely follows the pdf under consideration. By doing so, the probability $1/T$ of accepting Y increases, which causes the expected number of trials to decrease. To illustrate this, Fig. 2.6 gives an example of an acceptance-rejection algorithm for the Gamma$(k,1)$ distribution [391, 431]. The majoring function in this case is based on a Laplace distribution and is such that

$$\frac{\varphi(x)}{t(x)} = \left|\frac{(\theta-1)x}{\theta(k-1)}\right|^{k-1} \exp\left(-x + \frac{|x-(k-1)| + (k-1)(\theta+1)}{\theta}\right). \tag{2.4}$$

The Laplace distribution with location parameter $k-1$ and scale parameter θ — also called *double exponential* — is described by the pdf [391]

$$r(x) = \frac{1}{2\theta} \exp\left(\frac{|x-(k-1)|}{\theta}\right). \tag{2.5}$$

The alternative name double exponential comes from the fact that, when $k = 1$, for $x > 0$ the pdf (2.5) is just a scaled exponential pdf, which is reflected around the y-axis to get the $x < 0$ part. The pdf (2.5) is simple enough that we can easily use inversion to perform Step 1 of the algorithm described in Fig. 2.6; see Prob. 2.10.

1. Generate a Laplace variate Y with location parameter $k-1$ and scale $\theta = 1 + \sqrt{4k-3}/2$.
2. If $Y < 0$, then return to Step 1.
3. $U \leftarrow$ `Rand01()`.
4. If $U \leq \varphi(Y)/t(Y)$, then return Y; otherwise go back to Step 1.

Fig. 2.6 Steps describing an acceptance-rejection algorithm for the gamma distribution with parameters $(k, 1)$, where $\varphi(\cdot)/t(\cdot)$ is given in (2.4). At least two uniform numbers are used every time we go through these four steps.

2.4 Composition

This method can be used when the CDF from which we want to generate observations can be written as a sum,

$$F(x) = \sum_{i=1}^{\infty} p_i F_i(x), \tag{2.6}$$

where $p_i \geq 0$, $\sum_{i=1}^{\infty} p_i = 1$, and each $F_i(\cdot)$ is a CDF. Hence a random variable with a CDF of the form (2.6) is such that with probability p_i it has a distribution determined by $F_i(\cdot)$. We can then use the algorithm shown in Fig. 2.7 to generate variates from a CDF of the form (2.6).

Of course, each of the two steps themselves require that some generating method be used, for instance inversion based on two independent uniform numbers U_1 and U_2 (one for generating I, the other for X). Note also that,

1. Generate I according to $P(I = i) = p_i$.
2. Return an observation X having CDF $F_I(\cdot)$ and independent from I.

Fig. 2.7 Steps describing how to use composition to generate random variates.

unlike inversion, we need at least two uniform numbers to generate one variate.

The composition method arises naturally for mixture distributions, but it can also be useful for tackling complicated density functions by breaking them down into different components, in which case p_i corresponds to the area under the curve of the ith component. We illustrate this idea in Example 2.2.

Example 2.2. Consider the beta density function $\varphi(x) = 12x^2(1-x)$ for $0 \leq x \leq 1$. Here we can form a piecewise linear function as illustrated in Fig. 2.8. This function passes through the maximum of $\varphi(x)$ occurring at $(2/3, 16/9)$; the inflection point $(1/3, 8/9)$, where the second derivative of $\varphi(x)$ becomes negative; the endpoint (1,0); and the point $(1/9, 0)$ obtained by drawing a line from the inflection point $(1/3, 8/9)$ that has the same slope as $\varphi(x)$ at that point. (This slope is given by 4.) The remainder of the area under the curve of $\varphi(x)$ can then be split into three areas. The area under the curve of the piecewise linear function can be shown to be 68/81, which means that about 84% of the draws based on the composition method will require generating observations from a distribution with a piecewise linear pdf, something that is relatively easy to achieve (see Prob. 2.6). Problem 2.5 at the end of the chapter asks you to find the corresponding values of p_i and $F_i(x)$, $i = 1, \ldots, 4$, for Fig. 2.8.

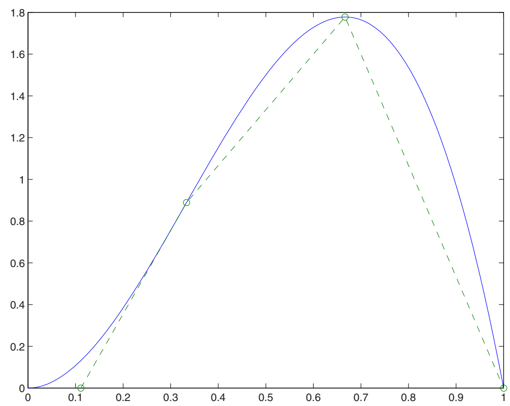

Fig. 2.8 Composition applied to the beta pdf $12x^2(1-x)$. The area under the curve is partitioned into four pieces.

2.5 Convolution and other useful identities

The convolution method is useful for random variables that can be written as a sum of i.i.d. random variables, typically coming from a simpler distribution. More precisely, we assume $X = Y_1 + \ldots + Y_n$, where the Y_i are i.i.d. random variables. Well-known examples are as follows:

1. $X \sim \mathrm{Gamma}(n, \beta)$: $Y_i \sim \mathrm{Exp}(\beta)$.
2. $X \sim \chi^2(n)$: $Y_i = Z_i^2$, where $Z_i \sim N(0, 1)$.
3. $X \sim \mathrm{Binomial}(n, p)$: $Y_i \sim \mathrm{Bernoulli}(p)$.
4. $X \sim$ Negative Binomial(n, p): $Y_i \sim \mathrm{Geometric}(p)$.

The main disadvantage of this method is that it requires that n random variates be generated in order to get a single observation from X.

More generally, relationships between different distributions can be used for random variate generation. For instance, Fox [126] uses the fact that, for a sample of n i.i.d. uniform variates in $[0, 1]$, the ith order statistic has a beta distribution with parameters $(i, n + 1 - i)$. Based on this, he suggests the method shown in Fig. 2.9 for generating a random variate $X \sim \mathrm{Beta}(a, b)$, where a and b are positive integers.

1. Generate $a + b - 1$ i.i.d. uniform numbers in $(0, 1)$.
2. Return the ath smallest observation.

Fig. 2.9 Steps for generating a beta variate with parameters (a, b) using ranked data.

Another way of generating a beta variate is to use the fact that if Y_1 is a $\mathrm{Gamma}(a, 1)$ and Y_2 is a $\mathrm{Gamma}(b, 1)$, independent from Y_1, then $Y_1/(Y_1 + Y_2)$ is a $\mathrm{Beta}(a, b)$.

Finally, a clever way of generating normal variates, due to Box and Muller [36], exploits the idea that the joint pdf of two independent standard normal variables x and y is given by

$$\varphi_{X,Y}(x, y) = \frac{1}{2\pi} e^{-(x^2+y^2)/2}, -\infty < x, y < \infty.$$

We can then perform a change of variables using polar coordinates — which is why a variation of this method, due to Marsaglia [301] and based on rejection, is called the *polar method* — as follows: $r = \sqrt{x^2 + y^2}$ and $\theta = \arctan(y/x)$. Hence we have $x = r\cos\theta$ and $y = r\sin\theta$, and the joint pdf of r and θ is

$$\varphi_{R,\Theta}(r, \theta) = \frac{|J|}{2\pi} e^{-r^2/2}, r > 0, 0 \leq \theta \leq 2\pi,$$

where $|J|$ is the Jacobian of the transformation given by

$$\begin{vmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{vmatrix} = r\cos^2\theta + r\sin^2\theta = r.$$

Hence $\varphi_{R,\Theta}(r,\theta) = (r/2\pi)e^{-r^2/2}$ with corresponding CDF

$$F_{R,\Theta}(r,\theta) = (\theta/2\pi)(1 - e^{-r^2/2}), r > 0, 0 \le \theta \le 2\pi.$$

Thus r and θ are independent, and we can generate them by inversion as

$$r = \sqrt{-\ln 2(1 - U_1)},$$
$$\theta = 2\pi U_2.$$

Transforming these back into x and y gives us the Box-Muller method described in Fig. 2.10. This method is quite popular for generating normal variates, but users should know that the sample produced when the source of randomness is a simple LCG has abnormal properties, as is illustrated nicely in [314].

$U_1 \leftarrow$ `Rand01()`
$U_2 \leftarrow$ `Rand01()`
$X_1 \leftarrow \sqrt{-2\ln(1-U_1)}\cos(2\pi U_2)$
$X_2 \leftarrow \sqrt{-2\ln(1-U_1)}\sin(2\pi U_2)$
return (X_1, X_2)

Fig. 2.10 Pseudocode for the Box-Muller method. It returns two independent standard normal variates.

2.6 Multivariate case

Here we consider the problem of generating vectors $(x_1, \ldots, x_k)$ of observations with a joint CDF $F(x_1, \ldots, x_k)$. First, a general approach that can be used is what we could call *nested conditioning* [243], where we generate each variate $x_1, \ldots, x_k$ successively, starting with x_1, for which we need the marginal distribution $F_{X_1}(x)$ given by

$$F_{X_1}(x) = \int_{-\infty}^{x}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\varphi(x_1, \ldots, x_k)dx_k \ldots dx_2 dx_1,$$

where $\varphi(x_1, \ldots, x_k)$ is the joint pdf associated with the CDF F. Once we have x_1, then we generate x_2 *conditionally* on x_1. That is, we generate an observation x_2 from $F_{X_2|X_1}(x|x_1)$ given by

$$F_{X_2|X_1}(x|x_1) = \int_{-\infty}^{x} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \frac{\varphi(x_1, \ldots, x_k)}{\varphi_1(x_1)} dx_k \ldots dx_3 dx_2,$$

where $\varphi_1(x_1)$ is the marginal pdf of X_1. We continue like this until the last variate x_k, generated from the conditional distribution

$$F_{X_k|X_1,\ldots,X_{k-1}}(x|x_1, \ldots, x_{k-1}).$$

Of course, for this method to be applicable, we need to be able to determine the marginal and conditional distributions and have a way of generating variates from each of them. Also, the efficiency of the method depends heavily on the order we chose for generating the variates x_i. That is, among the $k!$ possible choices, some might lead to a much faster generation of the vector $(x_1, \ldots, x_k)$ [391].

Here is a simple example to illustrate this method.

Example 2.3. Suppose we want to generate a vector (x_1, x_2) having the joint pdf

$$\varphi(x_1, x_2) = \begin{cases} 2 & \text{if } 0 \le x_2 \le x_1 \le 1 \\ 0 & \text{else.} \end{cases} \tag{2.7}$$

We have that the marginal pdf of X_1 is

$$\varphi_1(x_1) = \int_0^{x_1} 2dx_2 = 2x_1, \qquad 0 \le x_1 \le 1,$$

and thus the marginal CDF of X_1 is

$$F_{X_1}(x_1) = x_1^2, \qquad 0 \le x_1 \le 1.$$

We must then get the conditional pdf of X_2 given X_1,

$$\varphi_{X_2|X_1}(x_2|x_1) = \frac{1}{x_1}, \qquad 0 \le x_2 \le x_1 \le 1,$$

so that the conditional CDF of X_2 given $X_1 = x_1$ is

$$F_{X_2|X_1}(x_2|x_1) = \frac{x_2}{x_1}, \qquad 0 \le x_2 \le x_1.$$

Overall, the algorithm shown in Fig. 2.11 can be used to generate (x_1, x_2).

Second, an important case to discuss is the multinormal distribution. That is, suppose we want to generate a vector $(x_1, \ldots, x_k)$ that follows a multinormal distribution with mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)^{\mathrm{T}}$ and covariance matrix Σ. In that case, we can use the fact that if $\mathbf{Z} = (Z_1, \ldots, Z_k)^{\mathrm{T}}$ is a vector of i.i.d. standard normal random variables, then $A\mathbf{Z}$ has a multinormal distribution with mean zero and covariance matrix AA^{T}. Hence, by using a matrix C such that $CC^{\mathrm{T}} = \Sigma$, we can use the identity

```
U_1 ← Rand01()
x_1 ← √U_1
U_2 ← Rand01()
x_2 ← x_1 U_2
return(x_1, x_2)
```

$$
\begin{aligned}
&U_1 \leftarrow \texttt{Rand01()}\\
&x_1 \leftarrow \sqrt{U_1}\\
&U_2 \leftarrow \texttt{Rand01()}\\
&x_2 \leftarrow x_1 U_2\\
&\mathrm{return}(x_1, x_2)
\end{aligned}
$$

Fig. 2.11 Pseudocode for using nested conditioning for the simple bivariate distribution (2.7).

$$\mathbf{X} = \boldsymbol{\mu} + C\mathbf{Z},$$

where $\mathbf{X} = (x_1, \ldots, x_k)^{\mathrm{T}}$. To get a matrix C such that $CC^{\mathrm{T}} = \Sigma$, we can use the lower-triangular matrix obtained from the Cholesky decomposition of Σ. As we will see in Chap. 6, other choices might be more suitable when using quasi–Monte Carlo sampling.

The third case we discuss is the use of copulas to model a joint distribution. The general approach to generate a vector $(x_1, \ldots, x_k)$ of variates having the joint CDF $F(x_1, \ldots, x_k)$ given by (2.1) is shown in Fig. 2.12.

Generate $(u_1, \ldots, u_k)$ according to C.
return $x_j = H_j^{-1}(u_j)$, $j = 1, \ldots, k$.

Fig. 2.12 Steps describing the general approach for generating random variates modeled using a copula C and having marginal CDF $H_1, \ldots, H_k$.

We illustrate with the following two examples how models described by copulas tend to lend themselves nicely to the use of inversion. More examples are given in [130, 462], for instance.

Example 2.4. Consider a bivariate *Gaussian copula*. In this case, we have $C(u, v) = \Phi_{2,\rho}(\Phi^{-1}(u), \Phi^{-1}(v))$, where Φ^{-1} denotes the inverse standard normal CDF, and $\Phi_{2,\rho}$ represents the CDF of a bivariate normal with correlation coefficient ρ, for which the covariance matrix is

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

Here, we can generate (U_1, U_2) so that they follow C by first generating a vector (Z_1, Z_2) from the bivariate normal with correlation ρ and then set $U_1 = \Phi(Z_1)$ and $U_2 = \Phi(Z_2)$. This works since then we have

$$
\begin{aligned}
P(U_1 \le u_1, U_2 \le u_2) &= P(\Phi(Z_1) \le u_1, \Phi(Z_2) \le u_2) \\
&= P(Z_1 \le \Phi^{-1}(u_1), Z_2 \le \Phi^{-1}(u_2)) \\
&= \Phi_{2,\rho}(\Phi^{-1}(u_1), \Phi^{-1}(u_2)) = C(u_1, u_2).
\end{aligned}
$$

Note that the second equality in the display above holds because the inverse transform Φ^{-1} is a continuous and monotonically increasing function. Once we have (U_1, U_2) with the desired dependence structure — as prescribed by the copula — then we get X_1 and X_2 by applying the chosen marginal distribution to U_1 and U_2. That is, we let $X_1 = H_1^{-1}(U_1)$ and $X_2 = H_2^{-1}(U_2)$. This clearly produces a pair (X_1, X_2) with the correct distribution since

$$
\begin{aligned}
P(X_1 \le x_1, X_2 \le x_2) &= P(H_1^{-1}(U_1) \le x_1, H_2^{-1}(U_2) \le x_2) \\
&= P(U_1 \le H_1(x_1), U_2 \le H_2(x_2)) \\
&= C(H_1(x_1), H_2(x_2)).
\end{aligned}
$$

Example 2.5. A well-known family of copulas are the *Archimedean copulas*, which can be expressed as

$$
C(u_1, \ldots, u_d) = \phi^{-1}(\phi(u_1) + \ldots + \phi(u_k)),
$$

where ϕ is a convex, decreasing function with domain $(0, 1]$ and range $[0, \infty)$ such that $\phi(1) = 0$, and is called the *generator* of the copula. A member of this family is *Frank's bivariate copula*, where

$$
C(u_1, u_2) = \frac{1}{\alpha} \ln\left(1 + \frac{(\exp(\alpha u_1) - 1)(\exp(\alpha v_1) - 1)}{\exp(\alpha) - 1}\right).
$$

For this special case, correlated uniform numbers (U_1, U_2) following this bivariate CDF can be generated as in Fig. 2.13, where $\tilde{\alpha} = e^{\alpha}$ [136].

FrankBivCopula($\tilde{\alpha}$)
  $V_1 \leftarrow$ `Rand01()`
  $V_2 \leftarrow$ `Rand01()`
  $T \leftarrow \tilde{\alpha}^{V_1} + (\tilde{\alpha} - \tilde{\alpha}^{V_1})V_2$
  $U_1 \leftarrow V_1$
  $U_2 \leftarrow \log_{\tilde{\alpha}}[T/(T + (1 - \tilde{\alpha})U_2)]$
  return (U_1, U_2)

Fig. 2.13 Pseudocode showing how to generate (U_1, U_2) according to Frank's bivariate copula.

Problems

2.1. Show that if $\{B(t), t \geq 0\}$ is a standard Brownian motion, then we have that $\mathrm{Cov}(B(s), B(t)) = \min(s, t)$ for $t, s \geq 0$.

2.2. A Weibull random variable has a pdf given by

$$\varphi(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k},$$

where $k > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter. Describe an algorithm that uses inversion to generate random variates having a Weibull distribution with generic parameters (k, λ).

2.3. Suppose you want to generate observations from a truncated distribution. That is, for some real numbers $a < b$ and some pdf $\varphi(x)$ (with associated CDF $F(\cdot)$), $\infty < x < \infty$, you want to generate random variates having the truncated pdf

$$\tilde{\varphi}(x) = \begin{cases} \frac{\varphi(x)}{F(b)-F(a)} & a \leq x \leq b \\ 0 & \text{else.} \end{cases}$$

Assume the inverse CDF $F^{-1}(\cdot)$ can be computed. Describe an algorithm to generate variates from the truncated pdf above.

2.4. Describe an algorithm to generate observations from the continuous empirical distribution $\tilde{F}_n$ defined in Prob. 1.15.

2.5. Compute the values of p_i and $F_i(x)$ for the composition method applied to the beta pdf $\varphi(x) = 12x^2(1 - x)$ discussed in Example 2.2.

2.6. Consider the pdf that corresponds to the piecewise linear function shown in Fig. 2.8, which, as discussed in Example 2.2, accounts for about 85% of the draws when using the composition method. (a) Give an expression for that pdf. (b) Give an algorithm to generate variates from this pdf using inversion.

2.7. For the beta pdf $\varphi(x) = 12x^2(1 - x)$, $0 \leq x \leq 1$, implement the acceptance-rejection approach described on p. 47, and for a sample of 100,000 beta variates compute the average number of uniform variates required to output one beta variate.

2.8. An example of an acceptance-rejection algorithm to generate random variates is given in [11, p. 25]. In this case, the goal is to generate three-dimensional random unit vectors. To do so by acceptance-rejection, the idea is to generate a random point uniformly in $[-1, 1)^3$, accept it if it is within the unit sphere centered at (0, 0, 0) (and then rescale it so that its length is one), and reject it otherwise. (a) Prove that this method correctly generates a random unit vector. (b) What is the expected number of trials required in order to generate one vector? (c) Use a two-dimensional version of that method to perform the *Buffon's needle* experiment, which can be used to estimate π as follows [42]. Throw n needles of length 0.5 on a floor with planks of width 1 and infinite length; estimate π by the fraction n/k, where k is the number of times the needle fell across a crack in the floor. To simplify things, assume we want to estimate $1/\pi$ and thus can use the approximation k/n. Use $n = 1000$, and verify whether a 95% confidence interval based on this sample contains $1/\pi$ or not.

2.9. Consider a random variable X having the following probability distribution:

$$\begin{aligned} P(X = 0) &= 0.05, \\ P(X = 1) &= 0.10, \\ P(X = 2) &= 0.15, \\ P(x < X \leq y) &= c(y - x) \text{ for } 0 < x < y < 1 \\ &\qquad\qquad \text{and } 1 < x < y < 2. \end{aligned}$$

(a) Find the value of c such that the distribution above is a valid probability distribution. (b) Give an algorithm using inversion to generate random variates having the distribution above. Make sure the transformation you use is monotone.

2.10. Consider the Laplace distribution whose pdf is given in (2.5). (a) Describe one way of applying composition to generate Laplace random variates. (b) Describe how to use inversion to generate Laplace random variates.

2.11. Consider the bivariate distribution under study in the pseudocode given in Fig. 2.11. Suppose the goal is to estimate $\mu = \mathrm{E}(X_1+X_2)$ by drawing n i.i.d. pairs of observations $(x_{i,1}, x_{i,2})$ for $i = 1, \ldots, n$. (a) Compute the variance of the estimator obtained based on the approach described in Fig. 2.11. (b) Give pseudocode for the approach that consists in first generating X_2 instead of X_1. (c) Compare the variance of the estimator for μ obtained using the approach in (b) with the one from (a).

2.12. Consider a multivariate normal vector $\mathbf{X}$ with covariance matrix Σ having entries of the form $\sigma_{ij} = \sigma_i \sigma_j \rho_{ij}$, where σ_i^2 is the variance of X_i, for $i = 1, \ldots, d$, and ρ_{ij} is the correlation between X_i and X_j for $1 \leq i, j \leq d$. Give a formula for the entries of the $d \times d$ lower-triangular matrix C obtained by Cholesky decomposition of Σ.

2.13. Find the generator ϕ corresponding to the *Gumbel-Hougaard copula* [130]

$$C(u, v) = \exp\left\{ - \left[(-\ln u)^{\alpha} + (-\ln v)^{\alpha}\right]^{1/\alpha} \right\}.$$

2.14. Show that the pair (U_1, U_2) output by the algorithm described in Fig. 2.13 has the desired distribution.

http://www.springer.com/978-0-387-78164-8

Monte Carlo and Quasi-Monte Carlo Sampling
Lemieux, C.
2009, XIV, 373 p., Hardcover
ISBN: 978-0-387-78164-8