4 Rejection Method

The inverse transform method works well when the cdf F of a random variable does not have a complicated expression and its inverse F^{-1} is relatively easy to find. But when that is not the case, we need other methods with larger applicability. We present next a method that uses the pdf f instead.

Remark 4.1. Before we get started, let us briefly review some properties of Uniformly distributed random variables and random vectors.

1. Recall that for Uniform $U(a, b)$ variables, the pdf is given by

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & x \in [a, b] \\ 0, & x \notin [a, b] \end{cases} = \begin{cases} \dfrac{1}{\text{length}[a,b]}, & x \in [a, b] \\ 0, & x \notin [a, b] \end{cases}.$$

A vector (X, Y) has a Uniform distribution over a domain $D \subseteq \mathbb{R}^2$ if its (joint) pdf is of the form

$$f(x, y) = \begin{cases} \text{const}, & (x, y) \in D \\ 0, & (x, y) \notin D \end{cases}.$$

What value must that constant be? It should be the value that makes the total integral of f over $\mathbb{R}^2$ equal to 1, as that represents the probability of the sure event. Then

$$1 = \iint_{\mathbb{R}^2} f(x, y)\, dxdy = \iint_D \text{const}\, dxdy = \text{const} \iint_D dxdy = \text{const} \cdot \text{area}(D),$$

so that constant must be $1/\text{area}(D)$. Thus, a vector (X, Y) has a Uniform distribution over $D \subseteq \mathbb{R}^2$ if its pdf is

$$f(x, y) = \begin{cases} \dfrac{1}{\text{area}(D)}, & (x, y) \in D \\ 0, & (x, y) \notin D \end{cases}. \tag{4.1}$$

2. From the joint pdf of a vector (X, Y), we can always get the *marginal* pdf's of its components by

$$f_X(x) = \int_{\mathbb{R}} f(x, y)\, dy, \ \forall x \in \mathbb{R}, \quad f_Y(y) = \int_{\mathbb{R}} f(x, y)\, dx, \ \forall y \in \mathbb{R}. \tag{4.2}$$

3. If $U \in U(0, 1)$, then $X = \alpha + (\beta - \alpha)U \in U(\alpha, \beta)$, $\forall \alpha < \beta$.

Theorem 4.2. *Let* $f : \mathbb{R} \to \mathbb{R}$ *be a pdf. Let the vector* (X, Y) *be Uniformly distributed over the region*

$$D = \{(x, y) \in \mathbb{R}^2 \mid 0 \le y \le f(x)\} \tag{4.3}$$

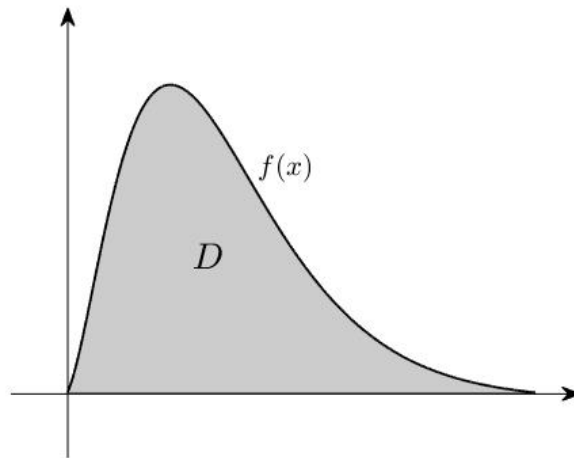*(see Figure 1). Then* X *has pdf* f*, i.e.* $f_X = f$.

Fig. 1: Domain D

Proof. First, let us determine the joint pdf of the vector (X, Y). By (4.1), it is

$$f_{(X,Y)}(x, y) = \frac{1}{\text{area}(D)}, \text{ for } (x, y) \in D$$

and 0 everywhere else. But, since f is a pdf, that area is $\displaystyle\int_{\mathbb{R}} f(x)\, dx = 1$.

So, the joint pdf of (X, Y) is

$$f_{(X,Y)}(x, y) = \begin{cases} 1, & (x, y) \in D \\ 0, & (x, y) \notin D. \end{cases}$$

Then, using (4.2), we find the (marginal) pdf of its first component X. Fix $x \in \mathbb{R}$. We have

$$f_X(x) = \int_{\mathbb{R}} f_{(X,Y)}(x, y)\, dy = \int_{(x,y) \in D} dy = \int_{y=0}^{y=f(x)} dy = f(x). \tag{4.4}$$

Thus, X indeed has the function f as its pdf. □

2

So, to generate a variable with given pdf f, we generate points (X, Y) that are Uniformly distributed in D. In order to have $(X, Y) \in D$, we must have $Y \leq f(X)$. If that is not the case, we *reject* the value, hence the name of the method.

Algorithm 4.3.

1. Find numbers $a, b \in \mathbb{R}, c \in \mathbb{R}_+$ such that $f(x) \in [0, c]$ for $x \in [a, b]$ (this is always possible, since D is a bounded region in $\mathbb{R}^2$, having an area of 1). The rectangle $[a, b] \times [0, c]$ is called a *bounding box*.

2. Generate $U, V \in U(0, 1)$.

3. Let $X = a + (b - a)U$ and $Y = cV$. Then $X \in U(a, b)$, $Y \in U(0, c)$ and $(X, Y) \in U([a, b] \times [0, c])$.

4. If $Y > f(X)$, reject the point and return to step 2. If $Y \leq f(X)$, then X has the desired pdf, f.

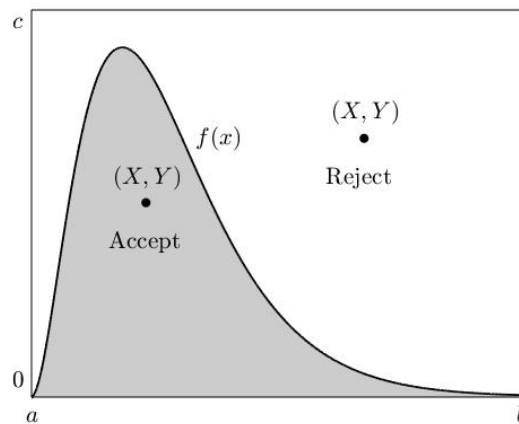The idea of the rejection method is displayed graphically in Figure 2.

Fig. 2: Rejection Method

Example 4.4. Recall Example 3.2 (Lecture 3). Now let us use the rejection method to generate a random variable X with that same pdf

$$f(x) = \frac{1}{2}(x + 1), \ x \in [-1, 1]. \tag{4.5}$$

Then test it for the values $(U_1, V_1) = (0.12, 0.45)$ and $(U_2, V_2) = (0.91, 0.37)$.

Solution. The graph of f is shown in Figure 3. We can see that a bounding box is $[-1, 1] \times [0, 1]$. Then by Algorithm 4.3, we find

$$
\begin{aligned}
X &= 2U - 1 \text{ and} \\
Y &= V.
\end{aligned}
$$

For (U_1, V_1), we find $(X, Y) = (-0.76, 0.45)$, for which $Y = 0.45 > 0.12 = f(X)$, so this point is *rejected*.

For (U_2, V_2), we have $(X, Y) = (0.82, 0.37)$ and $Y = 0.37 < 0.91 = f(X)$, so this point is *accepted*.

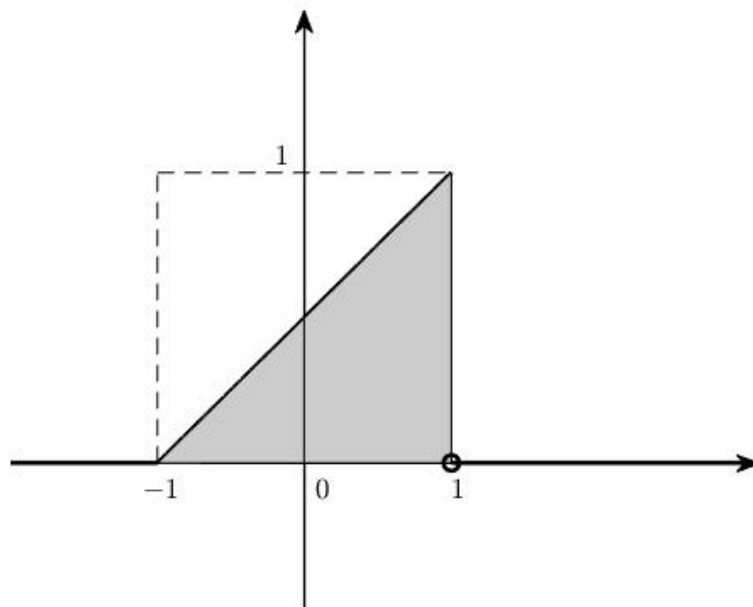Thus we generated the value 0.82 for the random variable X with pdf (4.5).

Fig. 3: Function f in Example 4.4

■

Remark 4.5. The rejection method can be used to generate $n-$dimensional random vectors having a desired joint pdf $f : \mathbb{R}^n \to \mathbb{R}$. A bounding box now becomes an $(n+1)-$dimensional cube, where we generate a Uniformly distributed random vector $(X_1, X_2, \ldots, X_n, Y)$, which will be accepted only if $Y \leq f(X_1, X_2, \ldots, X_n)$. Then, the generated vector $(X_1, X_2, \ldots, X_n)$ will have the desired joint density f.

5 Special Methods

These are methods aimed at particular variables, using specific properties of certain distributions. They are a good alternative of simulation, when the more general methods presented so far, are too complicated to implement.

Poisson Distribution $\mathcal{P}(\lambda)$, $\lambda > 0$

Let us recall: In a Poisson process, where X is the number of rare events occurring in time t, X has a Poisson distribution, $X \in \mathcal{P}(\lambda t)$, while the time between rare events and the time of the occurrence of the first rare event have an $Exp(\lambda)$ distribution. Then to generate a $\mathcal{P}(\lambda)$ variable, we count the number of rare events that occur during *one unit* of time ($t = 1$) and generate the Exponential times between events by formula (3.4) (Lecture 3), using the Inverse Transform Method. So, each such time is generated by $T_i = -\frac{1}{\lambda} \ln(U_i)$, for $U_i \in U(0,1)$ and then we count the number of events that occurred in one unit of time:

$$X = \max\{n \mid T_1 + \ldots + T_n \leq 1\}. \tag{5.1}$$

We can simplify the above formula:

$$
\begin{aligned}
T_1 + \ldots + T_n &= -\frac{1}{\lambda} \ln(U_1) + \ldots - \frac{1}{\lambda} \ln(U_n) \\
&= -\frac{1}{\lambda} \Big(\ln(U_1) + \ldots + \ln(U_n) \Big) \\
&= -\frac{1}{\lambda} \ln\big(U_1 \cdot \ldots \cdot U_n \big).
\end{aligned}
$$

Then in (5.1) we have, equivalently,

$$
\begin{aligned}
-\frac{1}{\lambda} \ln\big(U_1 \cdot \ldots \cdot U_n \big) &\leq 1 \\
\ln\big(U_1 \cdot \ldots \cdot U_n \big) &\geq -\lambda \\
U_1 \cdot \ldots \cdot U_n &\geq e^{-\lambda}.
\end{aligned}
$$

So, a Poisson variable X can be generated by

$$X = \max\{n \mid U_1 \cdot \ldots \cdot U_n \geq e^{-\lambda}\}. \tag{5.2}$$

Algorithm 5.1.

1. Generate $U_1, U_2, \ldots \in U(0, 1)$.

2. $X = \max\{n \mid U_1 \cdot U_2 \cdot \ldots \cdot U_n \geq e^{-\lambda}\}$.

Normal Distribution $N(\mu, \sigma)$, $\mu \in \mathbb{R}, \sigma > 0$

We present an algorithm called the **Box-Muller transform**, that converts a pair of independent Standard Uniform variables (U, V) into a pair of independent Standard Normal variables (Z_1, Z_2).

Algorithm 5.2.

1. Generate $U, V \in U(0, 1)$.

2. Let

$$
\begin{aligned}
Z_1 &= \sqrt{-2\ln(U)} \cos(2\pi V), \\
Z_2 &= \sqrt{-2\ln(U)} \sin(2\pi V).
\end{aligned}
\tag{5.3}
$$

Then Z_1, Z_2 are independent $N(0, 1)$ random variables.

3. Let $X = \sigma Z + \mu$ (for either Z from above). Then $X \in N(\mu, \sigma)$.

Without going into too many details, this transform is based on the following idea:
For $Z_1, Z_2 \in N(0, 1)$, the variable

$$
D^2 = Z_1^2 + Z_2^2 \in Exp(1/2),
$$

so D^2 can be generated by

$$
\begin{aligned}
D^2 &= -\frac{1}{1/2}\ln(U) = -2\ln(U) \text{ and} \\
D &= \sqrt{-2\ln(U)},
\end{aligned}
$$

for some $U \in U(0, 1)$. From here, it is just a matter of getting the two sides of a rectangular triangle from its hypotenuse (see Figure 4). We have

$$
\begin{aligned}
Z_1 &= D\cos\omega \\
Z_2 &= D\sin\omega,
\end{aligned}
$$

6

where the angle ω can take any value in $[0, 2\pi]$ (a complete rotation), i.e. it has a Uniform distribution $U(0, 2\pi)$, so $\omega = 2\pi V$ for some other Standard Uniform variable $V \in U(0, 1)$. Thus, we get (5.3).
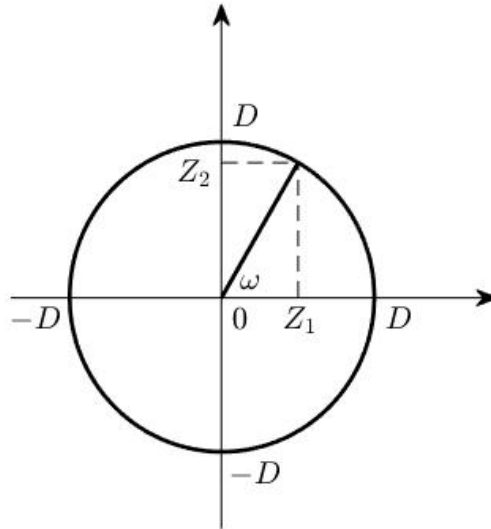

Fig. 4: Box-Muller Transform

6 Accuracy and Size of a Monte Carlo Study

Now, using the methods of simulation presented so far, we perform a Monte Carlo study, meaning that we put the chosen algorithm in a loop and simulate a "long run", i.e. generate a number of such variables, $X_1, \ldots, X_N$.

Recall from Statistics that when a parameter θ is approximated by an estimator (a function of sample variables) $\overline{\theta}$, a desired quality of that estimator is to be *unbiased*, i.e. that

$$E(\overline{\theta}) = \theta, \tag{6.1}$$

so that, in the long run, we know its values will stabilize at the right point. We also want that its variance $V(\overline{\theta})$ be small, approaching 0, as the sample size $N \to \infty$.

Estimating Probabilities, Means and Variances

We estimate probabilities by long run relative frequencies. For a random variable X, we generate variables $X_1, \ldots, X_N$ with the same distribution and approximate $p = P(X \in A)$ by

$$\bar{p} = \frac{\text{number of } X_1, \ldots, X_N \in A}{N}. \tag{6.2}$$

The mean value $E(X) = \mu$, the variance $V(X) = \sigma^2$ and the standard deviation $\sigma = \sqrt{V(X)}$ of a random variable X are estimated by

$$\begin{aligned}
\overline{X} &= \frac{X_1 + \ldots + X_N}{N}, \\
s^2 &= \frac{1}{N-1} \sum_{i=1}^{N} \left(X_i - \overline{X} \right)^2, \quad s = \sqrt{s^2},
\end{aligned} \tag{6.3}$$

respectively. Since the simulations are independent, the number at the numerator in (6.2) has Binomial $B(N, p)$ distribution (the number of successes in N trials) and, hence, expected value Np and variance $Np(1 - p)$. Then, we have

$$\begin{aligned}
E(\bar{p}) &= \frac{1}{N} Np = p, \\
V(\bar{p}) &= \frac{1}{N^2} Np(1 - p) = \frac{p(1 - p)}{N}.
\end{aligned} \tag{6.4}$$

Thus, $\bar{p}$ is an unbiased estimator for p and its standard deviation $\sigma(\bar{p}) = \sqrt{\dfrac{p(1 - p)}{N}}$ decreases with N at the rate of $1/\sqrt{N}$.

The same is true for the estimators in (6.3), but we omit the details.

Accuracy of a Monte Carlo Study

When we conduct a Monte Carlo study, the question arises about its size. What would be a suitable size in order to get a certain accuracy? Given a tolerable error $\varepsilon > 0$ and a significance level (probability of error) $\alpha \in (0, 1)$, we want to determine the size N so that

$$P\left(|\bar{p} - p| > \varepsilon \right) \leq \alpha. \tag{6.5}$$

It is known that for moderate values of p ($0.05 \leq p \leq 0.95$) and large values of n, a Binomial variable can be approximated by a Normal one:

$$B(n, p) \approx N\left(\mu = np, \ \sigma = \sqrt{np(1-p)}\right) \tag{6.6}$$

Also, recall that for a Normal variable $X \in N(\mu, \sigma)$, its *reduced* variable $\dfrac{X - E(X)}{\sigma(X)} = \dfrac{X - \mu}{\sigma}$ has a Standard Normal $N(0, 1)$ distribution.

Then for the variable $N\bar{p}$ for large values of N, we use the Normal approximation of the Binomial distribution of $N\bar{p}$, to get

$$\frac{N\bar{p} - E(N\bar{p})}{\sqrt{V(N\bar{p})}} = \frac{N(\bar{p} - p)}{N\sqrt{\dfrac{p(1-p)}{N}}} = \frac{\bar{p} - p}{\sqrt{\dfrac{p(1-p)}{N}}} \approx N(0, 1).$$

We can use this to estimate the probability in (6.5). We have

$$P\left(|\bar{p} - p| > \varepsilon\right) = P\left(\frac{|\bar{p} - p|}{\sqrt{\dfrac{p(1-p)}{N}}} > \frac{\varepsilon}{\sqrt{\dfrac{p(1-p)}{N}}}\right) = 2\Phi\left(-\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}}\right),$$

where Φ is Laplace's function (the cdf of a $N(0, 1)$ variable) described in equation (5.13) (Lecture 2).

Still, this contains the unknown value p. We can manage that using the fact that for any $p \in (0, 1)$,

$$
\begin{aligned}
p(1-p) &\leq \frac{1}{4}, \\
\sqrt{p(1-p)} &\leq \frac{1}{2}, \\
\frac{1}{\sqrt{p(1-p)}} &\geq 2, \\
-\frac{1}{\sqrt{p(1-p)}} &\leq -2,
\end{aligned}
$$

so,

$$-\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}} \leq -2\varepsilon\sqrt{N}.$$

Since Φ is an increasing function,

$$\Phi\left(-\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}}\right) \leq \Phi\left(-2\varepsilon\sqrt{N}\right).$$

Then to ensure (6.5), we take $\Phi\left(-2\varepsilon\sqrt{N}\right) \leq \alpha/2$, or, equivalently, $-2\varepsilon\sqrt{N} \leq \Phi^{-1}(\alpha/2) = z_{\alpha/2}$, i.e.

$$N \geq \frac{1}{4}\left(\frac{z_{\alpha/2}}{\varepsilon}\right)^2, \tag{6.7}$$

where $z_{\alpha/2}$ is the quantile (inverse of the cdf Φ) of order $\alpha/2$ for the $N(0,1)$ distribution.

7   Other Applications of Monte Carlo Methods

Estimating lengths, areas, volumes

Let $A \subseteq [0,1]$. How to estimate length(A)? Recall that a variable $U \in U(0,1)$ has pdf $f_U(x) = 1$, $x \in [0,1]$. Then

$$P(U \in A) \;=\; \int_A f_U(x)\,dx \;=\; \int_A dx \;=\; \text{length}(A). \tag{7.1}$$

But MC methods can be used to estimate the probability on the left-hand side. So, we generate $U_1, \ldots, U_N \in U(0,1)$, compute the proportion of U_i that lie in A and estimate the length of A by that proportion.

What if $A \nsubseteq [0,1]$, but in some other interval $A \subseteq [a,b]$? Then for a variable $X \in U(a,b)$, with pdf $f(x) = 1/(b-a), x \in [a,b]$,

$$P(X \in A) \;=\; \int_A f(x)\,dx \;=\; \frac{1}{b-a}\int_A dx \;=\; \frac{1}{b-a}\text{length}(A). \tag{7.2}$$

Then we generate $X_1, \ldots, X_N \in U(a,b)$ and estimate the length of A by $(b-a)P(X \in A)$.

For estimating areas, we do exactly the same things, but with double integrals.

Let $A \subseteq [0,1] \times [0,1]$ and $U, V \in U(0,1)$. That means $(U,V) \in U\big([0,1] \times [0,1]\big)$, so its joint pdf is $f_{(U,V)}(x,y) = 1$, $(x,y) \in [0,1] \times [0,1]$. Then

$$P\big((U,V) \in A\big) \;=\; \iint_A f_{(U,V)}(x,y)\,dxdy \;=\; \iint_A dxdy \;=\; \text{area}(A). \tag{7.3}$$

Again, if $A \subseteq [a, b] \times [c, d]$ and $(X, Y) \in U([a, b] \times [c, d])$, then

$$P((X, Y) \in A) \ = \ \iint\limits_A f_{(X,Y)}(x, y) \, dxdy \ = \ \frac{1}{(b - a)(d - c)} \text{area}(A) \qquad (7.4)$$

and the area of A can be approximated by $(b - a)(d - c)P((X, Y) \in A)$.

Algorithm 7.1.

1. Generate $X_i \in U(a, b), Y_i \in U(c, d), i = 1, \ldots, N$.

2. Compute the number of pairs (X_i, Y_i) that belong to A, say N_A.

3. Estimate $\text{area}(A) \approx (b - a)(d - c)\dfrac{N_A}{N}$.

Example 7.2. Approximate π by MC methods.

Solution. The number π is the area of the unit disk $x^2 + y^2 \leq 1$. Cover the unit disk by the rectangle $[-1, 1] \times [-1, 1]$, i.e. find a bounding box (see Figure 5).
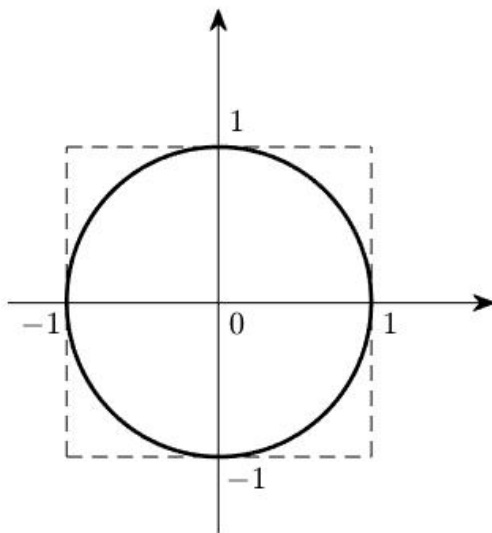

Fig. 5: Bounding box for the unit disk

Apply Algorithm 7.1:

1. Generate $X_1, \ldots, X_N, Y_1, \ldots, Y_N \in U(-1, 1)$.

2. Compute the number of pairs (X_i, Y_i) for which $X_i^2 + Y_i^2 \leq 1$, say N_π.

11

3. Approximate $\pi \approx 4\dfrac{N_\pi}{N}$.

■

Remark 7.3. Notice that estimation of areas (or lengths, volumes) by MC methods **does not require** knowing the *exact* boundaries of the region to be estimated. All that is necessary is covering that region by a bounding box (rectangle) and generating points Uniformly distributed in that rectangle.

Example 7.4. An emergency is reported at a nuclear power plant and it is necessary to assess the size of the region exposed to radioactivity. Boundaries of the region cannot be determined, but it is known that it is covered by a rectangle of 15 by 20 km and the level of radioactivity can be measured at any given location. Suppose that 100 measurements are taken at random points in that rectangular area and radioactivity is found above normal in 43 locations. Estimate the area of the exposed region A.

Solution. We have a bounding box $[0, 15] \times [0, 20]$ and the 100 locations represent 100 Uniform variables $(X_i, Y_i) \in U\Big([0, 15] \times [0, 20]\Big)$. Of those, 43 pairs are found to belong to the region A. Thus, we estimate
$$\text{area}(A) \approx 15 \cdot 20 \cdot \frac{43}{100} = 129 \text{ km}^2.$$

■

Monte Carlo integration

Recall that the definite integral of a nonnegative function represents the area of the region underneath the graph of that function. Then MC methods can be used to approximate definite integrals by estimating areas below or above the graphs of corresponding functions.

Suppose we want to approximate the integral

$$I \;=\; \int_a^b g(x)\, dx,$$

for some function $g : [a, b] \to [0, c]$. Then we cover the area that is I by $[a, b] \times [0, c]$ and estimate it with the rejection method.

Algorithm 7.5.

1. Generate $U_i, V_i \in U(0, 1), i = 1, \dots, N$.

2. Let $X_i = a + (b - a)U_i$ and $Y_i = cV_i, i = 1, \ldots, N$ (or generate directly $X_i \in U(a, b)$ and $Y_i \in U(0, c)$).

3. Compute the number of pairs (X_i, Y_i) for which $Y_i \leq g(X_i)$, say N_I.

4. Estimate the integral by

$$I \approx \bar{I} = (b - a)\, c\, \frac{N_I}{N}.$$

Finally, for the general case $g : [a, b] \rightarrow [c, d]$, where g can take both positive and negative values, take each subinterval separately and on those subintervals where $g(x) \leq 0$, consider $|g(x)|$. Then the areas above the x−axis are added and those below are subtracted (see Figure 6). Since the estimation of the integral still uses (long-run) proportions, the accuracy of the approximation is the same, i.e.

$$
\begin{aligned}
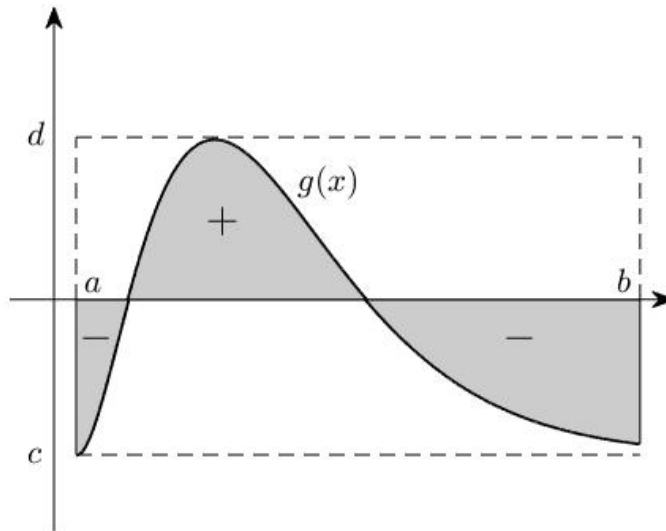E(\bar{I}) &= I \text{ (it is unbiased) and} \\
\sigma(\bar{I}) &= \sqrt{\frac{I(1 - I)}{N}}.
\end{aligned}
$$

Fig. 6: MC Integration