CHAPTER 9

VARIANCE REDUCTION

The estimation of performance measures in Monte Carlo simulation can be made more efficient by utilizing known information about the simulation model. The more that is known about the behavior of the system, the greater the amount of variance reduction that can be achieved. The main variance reduction techniques discussed in this chapter are:

1. Antithetic random variables.
2. Control variables.
3. Conditional Monte Carlo.
4. Stratification.
5. Latin hypercube sampling.
6. Importance sampling.
7. Quasi Monte Carlo. ☞ 25

For application of variance reduction techniques in rare-event simulation see Chapter 10. In particular, Section 10.6 and Chapter 14 both present a *splitting* approach to rare-event simulation. ☞ 381 ☞ 409 ☞ 481

9.1 VARIANCE REDUCTION EXAMPLE

Each of the variance reduction methods is illustrated using the following estimation problem concerning a bridge network. The problem is sufficiently complicated to warrant Monte Carlo simulation, while easy enough to implement, so that the workings of each technique can be concisely illustrated within the same context.

■ **EXAMPLE 9.1 (Bridge Network)**

Consider the undirected graph in Figure 9.1, depicting a **bridge network**.

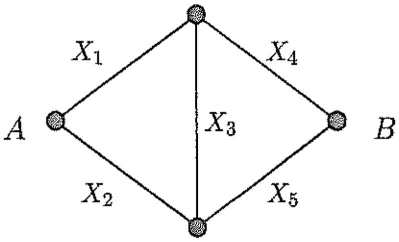


Figure 9.1 What is the expected length of the shortest path from A to B?

Suppose we wish to estimate the expected length ℓ of the shortest path between nodes (vertices) A and B, where the lengths of the links (edges) are random variables $X_1, \ldots, X_5$. We have $\ell = \mathbb{E}H(\mathbf{X})$, where

$$H(\mathbf{X}) = \min\{X_1 + X_4,\ X_1 + X_3 + X_5,\ X_2 + X_3 + X_4,\ X_2 + X_5\}\ . \tag{9.1}$$

Note that $H(\mathbf{x})$ is nondecreasing in each component of the vector $\mathbf{x}$. Suppose the lengths $\{X_i\}$ are independent and $X_i \sim \mathsf{U}(0, a_i)$, $i = 1, \ldots, 5$ with $(a_1, \ldots, a_5) = (1, 2, 3, 1, 2)$. Writing $X_i = a_i\, U_i$, $i = 1, \ldots, 5$ with $\{U_i\} \sim_{\text{iid}} \mathsf{U}(0, 1)$, we can restate the problem as the estimation of

$$\ell = \mathbb{E}h(\mathbf{U})\ , \tag{9.2}$$

where $\mathbf{U} = (U_1, \ldots, U_5)$ and $h(\mathbf{U}) = H(a_1 U_1, \ldots, a_5 U_5)$. The exact value can be determined by conditioning (see Section 9.4) and is given by

$$\ell = \frac{1339}{1440} = 0.9298611111\ldots\ .$$

☞ 306 *Crude Monte Carlo* (CMC) proceeds by generating $\mathbf{U}_1, \ldots, \mathbf{U}_N \overset{\text{iid}}{\sim} \mathsf{U}(0, 1)^5$ and returning

$$\widehat{\ell} = \frac{1}{N} \sum_{k=1}^{N} h(\mathbf{U}_k)$$

as an estimate for ℓ.

The following MATLAB program implements the CMC simulation. For a sample size of $N = 10^4$ a typical estimate is $\widehat{\ell} = 0.930$ with an estimated relative error of 0.43%.

```
%bridgeCMC.m
N = 10^4;
U = rand(N,5);
y = h(U);
est = mean(y)
percRE = std(y)/sqrt(N)/est*100
```

```
function out=h(u)
a=[1,2,3,1,2]; N = size(u,1);
X = u.*repmat(a,N,1);
Path_1=X(:,1)+X(:,4);
Path_2=X(:,1)+X(:,3)+X(:,5);
Path_3=X(:,2)+X(:,3)+X(:,4);
Path_4=X(:,2)+X(:,5);
out=min([Path_1,Path_2,Path_3,Path_4],[],2);
```

9.2 ANTITHETIC RANDOM VARIABLES

A pair of real-valued random variables (Y, Y^*) is called an **antithetic pair** if Y and Y^* have the same distribution and are *negatively correlated.* The main application of antithetic random variables in Monte Carlo estimation is based on the following theorem; see, for example, [18].

Theorem 9.2.1 (Antithetic Estimator) *Let N be an even number and let $(Y_1, Y_1^*), \ldots, (Y_{N/2}, Y_{N/2}^*)$ be independent antithetic pairs of random variables, where each Y_k and Y_k^* is distributed as Y. The* **antithetic estimator**

$$\widehat{\ell}^{(a)} = \frac{1}{N} \sum_{k=1}^{N/2} \{Y_k + Y_k^*\}, \tag{9.3}$$

is an unbiased estimator of $\ell = \mathbb{E}Y$, with variance

$$\begin{aligned}\mathrm{Var}(\widehat{\ell}^{(a)}) &= \frac{N/2}{N^2}\,(\mathrm{Var}(Y) + \mathrm{Var}(Y^*) + 2\,\mathrm{Cov}(Y, Y^*)) \\ &= (\mathrm{Var}(Y) + \mathrm{Cov}(Y, Y^*))/N \\ &= \frac{\mathrm{Var}(Y)}{N}\,(1 + \varrho_{Y,Y^*}),\end{aligned}$$

where ϱ_{Y,Y^*} is the correlation between Y and Y^*.

Note that (9.3) is simply the sample mean of the independent random variables $\{(Y_k + Y_k^*)/2\}$. Since the variance of the CMC estimator $\widehat{\ell} = N^{-1} \sum_{k=1}^{N} Y_i$ is $\mathrm{Var}(Y)/N$, the above theorem shows that the use of antithetic variables leads to a smaller variance of the estimator by a factor of $1 + \varrho_{Y,Y^*}$. The amount of reduction depends crucially on the amount of negative correlation between the antithetic variables.

In general, the output of a simulation run is of the form $Y = h(\mathbf{U})$, where h is a real-valued function and $\mathbf{U} = (U_1, U_2, \ldots)$ is a random vector of iid $\mathsf{U}(0,1)$ random variables. Suppose that $\mathbf{U}^*$ is another vector of iid $\mathsf{U}(0,1)$ random variables which is dependent on $\mathbf{U}$ and for which Y and $Y^* = h(\mathbf{U}^*)$ are negatively correlated. Then (Y, Y^*) is an antithetic pair. In particular, if h is a monotone function in each of its components, then the choice $\mathbf{U}^* = \mathbf{1} - \mathbf{U}$, where $\mathbf{1}$ is the vector of 1s, yields an antithetic pair.

☞ 306 An alternative to the CMC Algorithm 8.2 for estimating $\ell = \mathbb{E}Y = \mathbb{E}h(\mathbf{U})$ is thus as follows.

Algorithm 9.1 (Antithetic Estimation for Monotone h)

1. *Generate $Y_1 = h(\mathbf{U}_1), \ldots, Y_{N/2} = h(\mathbf{U}_{N/2})$ from independent simulation runs.*

2. *Let $Y_1^* = h(\mathbf{1} - \mathbf{U}_1), \ldots, Y_{N/2}^* = h(\mathbf{1} - \mathbf{U}_{N/2})$.*

3. *Compute the sample covariance matrix corresponding to the pairs $\{(Y_k, Y_k^*)\}$:*

$$C = \begin{pmatrix} \frac{1}{N/2-1}\sum_{k=1}^{N/2}(Y_k - \bar{Y})^2 & \frac{1}{N/2-1}\sum_{k=1}^{N/2}(Y_k - \bar{Y})(Y_k^* - \bar{Y}^*) \\ \frac{1}{N/2-1}\sum_{k=1}^{N/2}(Y_k - \bar{Y})(Y_k^* - \bar{Y}^*) & \frac{1}{N/2-1}\sum_{k=1}^{N/2}(Y_k^* - \bar{Y}^*)^2 \end{pmatrix}$$

4. *Estimate ℓ via the antithetic estimator $\widehat{\ell}^{(a)}$ in (9.3) and determine an approximate $1-\alpha$ confidence interval as*

$$\left(\widehat{\ell}^{(a)} - z_{1-\alpha/2}SE, \quad \widehat{\ell}^{(a)} + z_{1-\alpha/2}SE\right),$$

where SE is the estimated standard error:

$$SE = \sqrt{\frac{C_{1,1} + C_{2,2} + 2C_{1,2}}{2N}}\,,$$

and z_γ denotes the γ-quantile of the $\mathsf{N}(0,1)$ distribution.

For each of the $N/2$ runs in Step 2 one does not necessarily have to store the complete sequence $\mathbf{U} = (U_1, U_2, \ldots)$ of random numbers in memory, but simply
☞ 2 save the random seeds for each sequence.

■ EXAMPLE 9.2 (Antithetic Estimation for the Bridge Network)

The following MATLAB program implements an antithetic estimator of the expected length of the shortest path ℓ in Example 9.1. A typical estimate using $N = 10^4$ samples is $\widehat{\ell}^{(a)} = 0.929$ with an estimated relative error of 0.2%. Figure 9.2 illustrates that the correlation between $h(\mathbf{U})$ and $h(\mathbf{1} - \mathbf{U})$ is relatively high in this case. The correlation coefficient is around -0.77, which means a more than fourfold reduction in simulation effort when compared to CMC. Function `h.m` in the code that follows is the same as in Example 9.1.

```
%compare_CMC_and_ARV.m
N=10^4;
U=rand(N/2,5);  % get uniform random variables
y = h(U); ya = h(1-U);
ell=(mean(y) + mean(ya))/2;
C=cov(y,ya);
var_h = sum(sum(C))/(2*N);
corr = C(1,2)/sqrt(C(1,1)*C(2,2));
fprintf('ell= %g,  RE = %g,  corr = %g\n',ell,sqrt(var_h)/ell, corr)
plot(y,ya,'.')
U = rand(N,5);
yb = h(U);
var_hb = var(yb)/N;
ReB = sqrt(var_hb)/ell
```

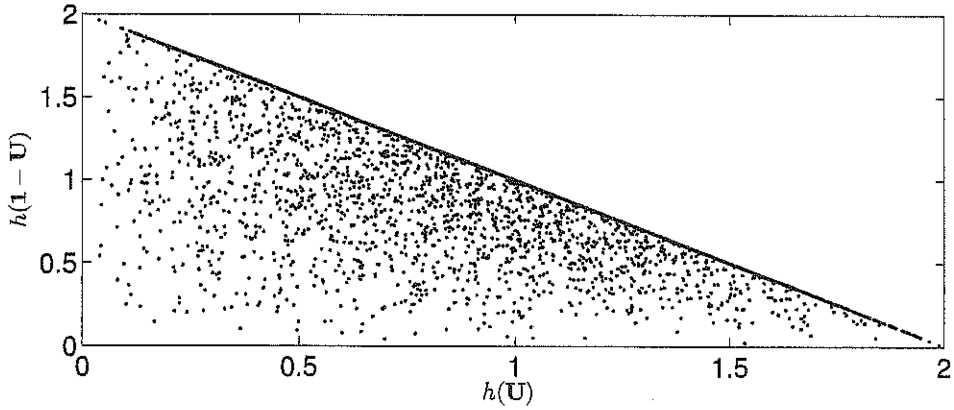


Figure 9.2 Scatter plot of $N = 10^4$ antithetic pairs (Y, Y^*) for the bridge network.

Remark 9.2.1 (Normal Antithetic Random Variables) Antithetic pairs can also be based on distributions other than the uniform. For example, suppose that $Y = H(\mathbf{Z})$, where $\mathbf{Z} = (Z_1, Z_2, \ldots)$ is a vector of iid standard normal random variables. By the inverse-transform method we can write $Y = h(\mathbf{U})$, with $h(\mathbf{u}) = H(\Phi^{-1}(u_1), \Phi^{-1}(u_2), \ldots)$, where Φ is the cdf of the $\mathsf{N}(0,1)$ distribution. Taking $\mathbf{U}^* = \mathbf{1} - \mathbf{U}$ gives $\mathbf{Z}^* = (\Phi^{-1}(U_1^*), \Phi^{-1}(U_2^*), \ldots) = -\mathbf{Z}$, so that (Y, Y^*) with $Y^* = H(-\mathbf{Z})$ forms an antithetic pair provided that Y and Y^* are negatively correlated, which is the case if H is a monotone function in each of its components. ☞ 45

9.3 CONTROL VARIABLES

Let Y be the output of a simulation run. A random variable $\widetilde{Y}$, obtained from the same simulation run, is called a **control variable** for Y if Y and $\widetilde{Y}$ are correlated (negatively or positively) and the expectation of $\widetilde{Y}$ is known. The use of control variables for variance reduction is based on the following observation.

Theorem 9.3.1 (Control Variable Estimation) *Let $Y_1, \ldots, Y_N$ be the output of N independent simulation runs, and let $\widetilde{Y}_1, \ldots, \widetilde{Y}_N$ be the corresponding control variables, with $\mathbb{E}\widetilde{Y}_k = \widetilde{\ell}$ known. Let $\varrho_{Y,\widetilde{Y}}$ be the correlation coefficient between each Y_k and $\widetilde{Y}_k$. For each $\alpha \in \mathbb{R}$ the (linear) estimator*

$$\widehat{\ell}^{(c)} = \frac{1}{N} \sum_{k=1}^{N} \left[Y_k - \alpha \left(\widetilde{Y}_k - \widetilde{\ell} \right) \right] \tag{9.4}$$

is an unbiased estimator for $\ell = \mathbb{E}Y$. The minimal variance of $\widehat{\ell}^{(c)}$ is

$$\mathrm{Var}(\widehat{\ell}^{(c)}) = \frac{1}{N} (1 - \varrho^2_{Y,\widetilde{Y}}) \mathrm{Var}(Y) \tag{9.5}$$

which is obtained for $\alpha = \mathrm{Cov}(Y, \widetilde{Y}) / \mathrm{Var}(\widetilde{Y})$.

Usually the optimal α in (9.5) is unknown, but it can be easily estimated from the sample covariance matrix of the $\{(Y_k, \widetilde{Y}_k)\}$. This leads to the following algorithm.

Algorithm 9.2 (Control Variable Estimation)

1. *From N independent simulation runs generate $Y_1, \ldots, Y_N$ and the control variables $\widetilde{Y}_1, \ldots, \widetilde{Y}_N$.*
2. *Compute the sample covariance matrix of $\{(Y_k, \widetilde{Y}_k)\}$:*

$$C = \begin{pmatrix} \frac{1}{N-1} \sum_{k=1}^{N} (Y_k - \bar{Y})^2 & \frac{1}{N-1} \sum_{k=1}^{N} (Y_k - \bar{Y})(\widetilde{Y}_k - \bar{\widetilde{Y}}) \\ \frac{1}{N-1} \sum_{k=1}^{N} (Y_k - \bar{Y})(\widetilde{Y}_k - \bar{\widetilde{Y}}) & \frac{1}{N-1} \sum_{k=1}^{N} (\widetilde{Y}_k - \bar{\widetilde{Y}})^2 \end{pmatrix} .$$

3. *Estimate ℓ via the control variable estimator $\widehat{\ell}^{(c)}$ in (9.4) with $\alpha = C_{1,2}/C_{2,2}$ and determine an approximate $1 - \alpha$ confidence interval as*

$$\left(\widehat{\ell}^{(c)} - z_{1-\alpha/2} SE, \quad \widehat{\ell}^{(c)} + z_{1-\alpha/2} SE \right),$$

where z_γ denotes the γ-quantile of the $\mathsf{N}(0,1)$ distribution and SE is the estimated standard error:

$$SE = \sqrt{\frac{1}{N} \left(1 - \frac{C_{1,2}^2}{C_{1,1} C_{2,2}} \right) C_{1,1}} \, .$$

■ EXAMPLE 9.3 (Control Variable Estimation for the Bridge Network)

Consider again the stochastic shortest path estimation problem for the bridge network in Example 9.1. As a control variable we can use, for example,

$$\widetilde{Y} = \min\{X_1 + X_4, X_2 + X_5\} .$$

This is particularly convenient for the current parameters $(1, 2, 3, 1, 2)$, as with high probability the shortest path will have a length equal to $\widetilde{Y}$; indeed, it will most likely have length $X_1 + X_4$, so that the latter would also be useful as a control variable. With a little calculation, the expectation of $\widetilde{Y}$ can be found to be $\mathbb{E}\widetilde{Y} = 15/16 = 0.9375$. Figure 9.3 shows the high correlation between the length of the shortest path $Y = H(\mathbf{X})$ defined in (9.1) and $\widetilde{Y}$. The corresponding correlation coefficient is around 0.98, which shows that a fiftyfold variance reduction in simulation effort is achieved compared with CMC estimation. The MATLAB program below implements the control variable estimator, using a sample size of $N = 10^4$. A typical estimate is $\widehat{\ell}^{(c)} = 0.92986$ with an estimated relative error of 0.05%. Function `h.m` in the code below is the same as in Example 9.1.

```
%bridgeCV.m
N=10^4;
u = rand(N,5);
Y = h(u);
Yc = hc(u);
plot(Y,Yc,'.')
C = cov(Y,Yc);
cor = C(1,2)/sqrt(C(1,1)*C(2,2))
alpha = C(1,2)/C(2,2);
yc = 15/16;
est = mean(Y - alpha*(Yc - yc))
RE = sqrt((1 - cor^2)*C(1,1)/N)/est
```

```
function out=hc(u)
a=[1,2,3,1,2];
N = size(u,1);
X = u.*repmat(a,N,1);
Path_1=X(:,1)+X(:,4);
Path_4=X(:,2)+X(:,5);
out=min([Path_1,Path_4],[],2);
```

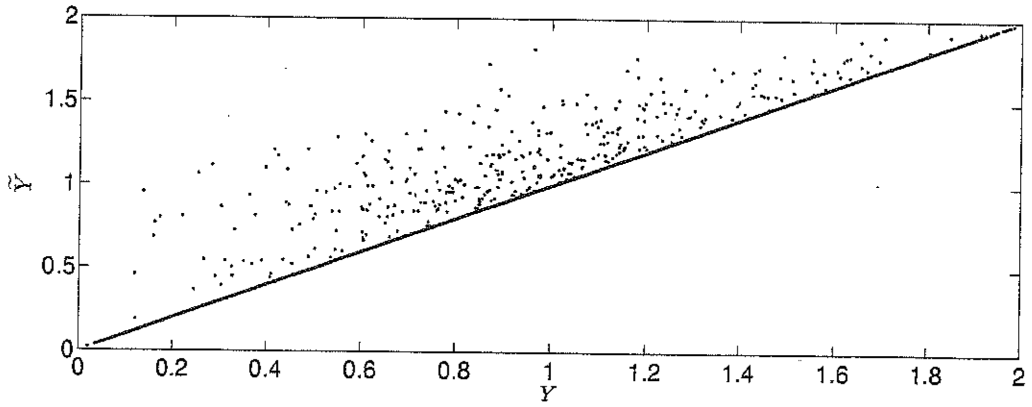


Figure 9.3 Scatter plot of $N = 10^4$ pairs $(Y, \widetilde{Y})$ for the output Y and control variable $\widetilde{Y}$ of the stochastic shortest path problem.

Remark 9.3.1 (Multiple Control Variables) Algorithm 9.2 can be extended straightforwardly to the case where more than one control variable is used for each output Y. Specifically, let $\widetilde{\mathbf{Y}} = (\widetilde{Y}_1, \ldots, \widetilde{Y}_m)^\top$ be a (column) vector of m control variables with known mean vector $\widetilde{\boldsymbol{\ell}} = \mathbb{E}\widetilde{\mathbf{Y}} = (\widetilde{\ell}_1, \ldots, \widetilde{\ell}_m)^\top$, where $\widetilde{\ell}_i = \mathbb{E}\widetilde{Y}_i$. Then, the control vector estimator of the optimal $\ell = \mathbb{E}Y$ based on independent random variables $Y_1, \ldots, Y_N$ with control vectors $\widetilde{\mathbf{Y}}_1 = (\widetilde{Y}_{11}, \ldots, \widetilde{Y}_{1m})^\top, \ldots, \widetilde{\mathbf{Y}}_N = (\widetilde{Y}_{N1}, \ldots, \widetilde{Y}_{Nm})^\top$ is given by

$$\widehat{\ell}^{(c)} = \frac{1}{N}\sum_{k=1}^{N}\left[Y_k - \boldsymbol{\alpha}^\top\left(\widetilde{\mathbf{Y}}_k - \widetilde{\boldsymbol{\ell}}\right)\right],$$

where $\boldsymbol{\alpha}$ is an estimator of the optimal vector $\boldsymbol{\alpha}^* = \Sigma_{\widetilde{\mathbf{Y}}}^{-1}\boldsymbol{\sigma}_{Y,\widetilde{\mathbf{Y}}}$. Here $\Sigma_{\widetilde{\mathbf{Y}}}$ is the $m \times m$ covariance matrix of $\widetilde{\mathbf{Y}}$, and $\boldsymbol{\sigma}_{Y,\widetilde{\mathbf{Y}}}$ is the $m \times 1$ vector whose i-th component is the covariance of Y and $\widetilde{Y}_i$, $i = 1, \ldots, m$. The variance of $\widehat{\ell}^{(c)}$ for $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*$ is

$$\mathrm{Var}(\widehat{\ell}^{(c)}) = \frac{1}{N}(1 - R^2_{Y,\widetilde{\mathbf{Y}}})\mathrm{Var}(Y)\,, \tag{9.6}$$

where

$$R^2_{Y,\widetilde{\mathbf{Y}}} = (\boldsymbol{\sigma}_{Y,\widetilde{\mathbf{Y}}})^\top\,\Sigma_{\widetilde{\mathbf{Y}}}^{-1}\,\boldsymbol{\sigma}_{Y,\widetilde{\mathbf{Y}}}/\mathrm{Var}(Y)$$

is the square of the **multiple correlation coefficient** of Y and $\widetilde{\mathbf{Y}}$. Again, the larger $R^2_{Y,\widetilde{\mathbf{Y}}}$ is, the greater the variance reduction.

9.4 CONDITIONAL MONTE CARLO

Variance reduction using **conditional Monte Carlo** is based on the following result.

Theorem 9.4.1 (Conditional Variance) *Let Y be a random variable and $\mathbf{Z}$ a random vector. Then*

$$\mathrm{Var}(Y) = \mathbb{E}\,\mathrm{Var}(Y\,|\,\mathbf{Z}) + \mathrm{Var}(\,\mathbb{E}[Y\,|\,\mathbf{Z}]\,)\,, \tag{9.7}$$

and hence $\mathrm{Var}(\mathbb{E}[Y\,|\,\mathbf{Z}]) \leqslant \mathrm{Var}(Y)$.

Suppose that the aim is to estimate $\ell = \mathbb{E}Y$, where Y is the output from a Monte Carlo experiment, and that one can find a random variable (or vector), $\mathbf{Z} \sim g$, such that the conditional expectation $\mathbb{E}[Y\,|\,\mathbf{Z} = \mathbf{z}]$ can be computed analytically. By the tower property (A.28), ☞ 618

$$\ell = \mathbb{E}Y = \mathbb{E}\,\mathbb{E}[Y\,|\,\mathbf{Z}]\,, \tag{9.8}$$

it follows that $\mathbb{E}[Y\,|\,\mathbf{Z}]$ is an unbiased estimator of ℓ. Moreover, by Theorem 9.4.1 the variance of $\mathbb{E}[Y\,|\,\mathbf{Z}]$ is always smaller than or equal to the variance of Y. The conditional Monte Carlo idea is sometimes referred to as **Rao–Blackwellization**.

Algorithm 9.3 (Conditional Monte Carlo)

1. *Generate a sample* $\mathbf{Z}_1, \ldots, \mathbf{Z}_N \overset{\text{iid}}{\sim} g$.
2. *Calculate* $\mathbb{E}[Y \,|\, \mathbf{Z}_k],\ k = 1, \ldots, N$ *analytically.*
3. *Estimate* $\ell = \mathbb{E}Y$ *by*

$$\widehat{\ell}_c = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}[Y \,|\, \mathbf{Z}_k] \tag{9.9}$$

and determine an approximate $1 - \alpha$ *confidence interval as*

$$\left(\widehat{\ell}_c - z_{1-\alpha/2} \frac{S}{\sqrt{N}},\quad \widehat{\ell}_c + z_{1-\alpha/2} \frac{S}{\sqrt{N}} \right),$$

where S *is the sample standard deviation of the* $\{\mathbb{E}[Y \,|\, \mathbf{Z}_k]\}$ *and* z_γ *denotes the* γ*-quantile of the* $\mathsf{N}(0, 1)$ *distribution.*

■ EXAMPLE 9.4 (Conditional Monte Carlo for the Bridge Network)

We return to Example 9.1. Let $Z_1 = \min\{X_4, X_3 + X_5\}$, $Z_2 = \min\{X_5, X_3 + X_4\}$, $Y_1 = X_1 + Z_1$, $Y_2 = X_2 + Z_2$, and $\mathbf{Z} = (Z_1, Z_2)$. Then, $Y = H(\mathbf{X})$ can be written as

$$Y = \min\{Y_1, Y_2\},$$

where conditional upon $\mathbf{Z} = \mathbf{z}$, (Y_1, Y_2) is uniformly distributed on the rectangle $\mathscr{R}_{\mathbf{z}} = [z_1, z_1 + 1] \times [z_2, z_2 + 2]$. The conditional expectation of Y given $\mathbf{Z} = \mathbf{z}$ can be evaluated exactly, and is given by

$$\mathbb{E}[Y \,|\, \mathbf{Z} = \mathbf{z}] = \begin{cases} \frac{1}{2} + z_1 & \text{if } \mathbf{z} \in \mathscr{A}_0\,, \\ \frac{5}{12} + \frac{3z_1}{4} - \frac{z_1^2}{4} - \frac{z_1^3}{12} + \frac{z_2}{4} + \frac{z_1 z_2}{2} + \frac{z_1^2 z_2}{4} - \frac{z_2^2}{4} - \frac{z_1 z_2^2}{4} + \frac{z_2^3}{12} & \text{if } \mathbf{z} \in \mathscr{A}_1\,, \\ \frac{1}{12}(5 - 3z_1^2 + 3z_2 - 3z_2^2 + z_1(9 + 6z_2)) & \text{if } \mathbf{z} \in \mathscr{A}_2\,, \end{cases}$$

where

$$\begin{aligned} \mathscr{A}_0 &= \{\mathbf{z} : 0 \leqslant z_1 \leqslant 1,\ z_1 + 1 \leqslant z_2 \leqslant 2\}, \\ \mathscr{A}_1 &= \{\mathbf{z} : 0 \leqslant z_1 \leqslant 1,\ z_1 \leqslant z_2 \leqslant z_1 + 1\}, \\ \mathscr{A}_2 &= \{\mathbf{z} : 0 \leqslant z_1 \leqslant 1,\ 0 \leqslant z_2 \leqslant z_1\}\,. \end{aligned}$$

For example, if $\mathbf{z} \in \mathscr{A}_1$, then the domain $\mathscr{R}_{\mathbf{z}}$ of (Y_1, Y_2) intersects the line $y_1 = y_2$ at $y_1 = z_2$ and $y_1 = z_1 + 1$, so that

$$\begin{aligned} \mathbb{E}[Y \,|\, \mathbf{Z} = \mathbf{z}] = &\int_{z_1}^{z_2} \int_{z_2}^{z_2+2} y_1 \, \frac{1}{2} \, \mathrm{d}y_2 \, \mathrm{d}y_1 + \int_{z_2}^{z_1+1} \int_{y_1}^{z_2+2} y_1 \, \frac{1}{2} \, \mathrm{d}y_2 \, \mathrm{d}y_1 \\ &+ \int_{z_2}^{z_1+1} \int_{z_2}^{y_1} y_2 \, \frac{1}{2} \, \mathrm{d}y_2 \, \mathrm{d}y_1 \,. \end{aligned}$$

The following MATLAB program gives an implementation of the corresponding conditional Monte Carlo estimator. A typical outcome for sample size $N = 10^4$ is

$\widehat{\ell}_c = 0.9282$ with an estimated relative error of 0.29%, compared with 0.43% for CMC, indicating more than a twofold reduction in simulation effort. Interestingly, the joint pdf of $\mathbf{Z}$ on $[0,1] \times [0,2]$ can, with considerable effort, be determined analytically, so that $\ell = \mathbb{E}Y$ can be evaluated exactly. This leads to the exact value given in the introduction:

$$\mathbb{E}Y = \frac{1339}{1440} = 0.9298611111\ldots .$$

```
%bridgeCondMC.m
N = 10^4;
S = zeros(N,1);
for i = 1:N
    u = rand(1,5);
    Z = Zcond(u);
    if Z(2)> Z(1) + 1
        S(i) = 1/2 + Z(1);
    elseif (Z(2) > Z(1))
        S(i) = 5/12 + (3*Z(1))/4 - Z(1)^2/4 - Z(1)^3/12 + Z(2)/4 ...
             + (Z(1)*Z(2))/2 + (Z(1)^2*Z(2))/4  ...
             -  Z(2)^2/4 - (Z(1)*Z(2)^2)/4 + Z(2)^3/12;
    else
        S(i) = (5 - 3*Z(1)^2 + 3*Z(2) - 3*Z(2)^2  ...
             + Z(1)*(9 + 6*Z(2)))/12;
    end
end
est = mean(S)
RE = std(S)/sqrt(N)/est
```

```
function Z=Zcond(u)
a=[1,2,3,1,2];
X = u*diag(a);
Z = [min([X(:,4), X(:,3) + X(:,5)],[],2),...
    min([X(:,5), X(:,3) + X(:,4)],[],2)];
```

9.5 STRATIFIED SAMPLING

Stratified sampling is closely related to both the composition method of Sec-
☞ 53 tion 3.1.2.6 and the conditional Monte Carlo method discussed in the previous
section. Let Y be the simulation output. The objective is to estimate $\ell = \mathbb{E}Y$.

Suppose that Y can be generated via the composition method. Thus, we assume that there exists a random variable Z taking values in $\{1, \ldots, m\}$, say, with known probabilities $\{p_i,\ i = 1, \ldots, m\}$. We further assume that it is easy to sample from the conditional distribution of Y given Z. The events $\{Z = i\}$, $i = 1, \ldots, m$,
partition the sample space Ω into disjoint **strata**; hence the name **stratification**.
☞ 618 Using the tower property (A.28), we can write

This representation suggests that we can estimate ℓ via the following **stratified sampling estimator**:

$$\widehat{\ell}^{(s)} = \sum_{i=1}^{m} p_i \frac{1}{N_i} \sum_{j=1}^{N_i} Y_{ij} \stackrel{\text{def}}{=} \sum_{i=1}^{m} p_i \, \bar{Y}_{i\bullet} \,, \tag{9.11}$$

where Y_{ij} is the j-th of N_i independent observations from the conditional distribution of Y given $Z = i$, $i = 1, \ldots, m$. How the strata should be chosen depends very much on the problem at hand. From (9.11), the variance of the stratified sampling estimator is given by

$$\mathrm{Var}(\widehat{\ell}^{(s)}) = \sum_{i=1}^{m} \frac{p_i^2 \, \sigma_i^2}{N_i} \,,$$

where $\sigma_i^2 = \mathrm{Var}(Y \mid Z = i)$ is the variance of Y within the i-th stratum, $i = 1, \ldots, m$.

For any given choice of the strata one can select the sample sizes $\{N_i\}$ in an optimal manner, as specified in the next theorem. A simple proof based on Lagrange multipliers can be found in [10].

Theorem 9.5.1 (Optimal Allocation of Sample Sizes) *The allocation of sample sizes (rounded to integers)*

$$N_i = N \, \frac{p_i \, \sigma_i}{\sum_{k=1}^{m} p_k \, \sigma_k} \,, \quad i = 1, \ldots, m \,, \tag{9.12}$$

where $\sigma_i^2 = \mathrm{Var}(Y \mid Z = i)$ provides the smallest variance for $\widehat{\ell}^{(s)}$ over all choices of $N_1, \ldots, N_m$ for which $\sum_i N_i = N$. The minimum value for the variance is

$$\frac{1}{N} \left(\sum_{i=1}^{m} p_i \, \sigma_i \right)^2 \,. \tag{9.13}$$

An obvious difficulty in applying Theorem 9.5.1 is that the standard deviations $\{\sigma_i\}$ are usually unknown. In practice, one can estimate the $\{\sigma_i\}$ from a pilot run and then proceed to estimate the optimal sample sizes from (9.12).

Algorithm 9.4 (Stratified Sampling)

1. *Choose the m strata and the sample sizes $N_i, i = 1, \ldots, m$ — the latter determined from (9.12) via a pilot run, for example.*

2. *For each stratum $i = 1, \ldots, m$ draw $Y_{i1}, \ldots, Y_{iN_i}$ independently from the conditional distribution of Y given $Z = i$, and let $\bar{Y}_{1\bullet}, \ldots, \bar{Y}_{m\bullet}$ be the sample means.*

3. *Estimate ℓ via (9.11), and calculate an approximate $1 - \alpha$ confidence interval as*

$$\left(\widehat{\ell}^{(s)} - z_{1-\alpha/2} \sqrt{\sum_{i=1}^{m} \frac{p_i^2 \, \widehat{\sigma_i^2}}{N_i}} \,, \quad \widehat{\ell}^{(s)} + z_{1-\alpha/2} \sqrt{\sum_{i=1}^{m} \frac{p_i^2 \, \widehat{\sigma_i^2}}{N_i}} \right) ,$$

where z_γ denotes the γ-quantile of the $\mathsf{N}(0,1)$ distribution and $\widehat{\sigma_i^2} = \sum_{j=1}^{N_i} (Y_{ij} - \bar{Y}_{i\bullet})^2 / (N_i - 1)$ is the sample variance of $\{Y_{ij}, j = 1, \ldots, N_i\}$.

If the sample size for the i-th stratum is chosen to be *proportional* to p_i, that is, $N_i = p_i\,N$ for some overall sample size N, then

$$\mathrm{Var}(\widehat{\ell}^{(s)}) = \sum_{i=1}^{m} \frac{p_i^2\,\sigma_i^2}{N_i} = \frac{1}{N}\,\mathbb{E}\,\mathrm{Var}(Y \,|\, Z) \leqslant \frac{1}{N}\mathrm{Var}(Y)\,, \tag{9.14}$$

so that the stratified estimator in this case (and hence under the optimal choice (9.12)) has a variance at least as small as the variance of the CMC estimator. This is called **proportional stratified sampling**. **Systematic sampling** [3] is proportional stratified sampling with equal weights; that is, $p_i = 1/m$ and $N_i = N/m = n$. The estimator (9.11) then reduces to

$$\widehat{\ell}^{(s)} = \frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{n} Y_{ij} = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{1}{m}\sum_{i=1}^{m} Y_{ij}\right) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{j=1}^{n} \bar{Y}_{\bullet j}\,. \tag{9.15}$$

Note that the $\{\bar{Y}_{\bullet j}\}$ are iid random variables, so that the standard error of $\widehat{\ell}^{(s)}$ can simply be estimated as $S/\sqrt{n}$, where S is the sample standard deviation of $\{\bar{Y}_{\bullet j}\}$.

Systematic sampling is especially useful when dealing directly with uniform random variables. Specifically, let $Y = h(U)$, where $U \sim \mathsf{U}(0,1)$, and define $Z = \lceil mU \rceil$ for some fixed $m \in \{1, 2, \ldots\}$. The events $\{Z = i\} = \{(i-1)/m \leqslant U < i/m\}$, $i = 1, \ldots, m$ divide the sample space into m equiprobable strata. Sampling Y conditional on $(i-1)/m \leqslant U < i/m$ is immediate. The systematic sampling procedure for the d-dimensional case is summarized in the following algorithm.

Algorithm 9.5 (Systematic Sampling for the d-Dimensional Hypercube) *Let $\ell = \mathbb{E}h(\mathbf{U})$, where $\mathbf{U} \sim \mathsf{U}(0,1)^d$. Suppose that the k-th component of the hypercube $(0,1)^d$ is divided up into K_k equal-length intervals, $k = 1, \ldots, d$, so that $(0,1)^d$ is divided into $m = \prod_{k=1}^{d} K_k$ hyperrectangles (ignoring the boundaries)*

$$\prod_{k=1}^{d}\left(\frac{i_k}{K_k}, \frac{i_k+1}{K_k}\right), \quad (i_1, \ldots, i_d) \in \mathscr{W}\,,$$

where $\mathscr{W} = \{(i_1, \ldots, i_d) : i_k \in \{0, 1 \ldots, K_k - 1\},\ k \in \{1, \ldots, d\}\}$.

1. *For each $\mathbf{i} = (i_1, \ldots, i_d) \in \mathscr{W}$ generate $\mathbf{V}_1, \ldots, \mathbf{V}_n \stackrel{\text{iid}}{\sim} \mathsf{U}(0,1)^d$ and evaluate*

$$Y_{\mathbf{i}j} = h\left(\frac{i_1 + V_{j1}}{K_1}, \ldots, \frac{i_d + V_{jd}}{K_d}\right), \quad j = 1, \ldots, n\,.$$

   Let

$$\bar{Y}_{\bullet j} = \frac{1}{m}\sum_{\mathbf{i} \in \mathscr{W}} Y_{\mathbf{i}j}\,.$$

2. *Estimate ℓ via (9.15), which is the sample mean of $\bar{Y}_{\bullet 1}, \ldots, \bar{Y}_{\bullet n}$, and determine an approximate $1-\alpha$ confidence interval as*

$$\left(\widehat{\ell}^{(s)} - z_{1-\alpha/2}\frac{S}{\sqrt{n}},\ \widehat{\ell}^{(s)} + z_{1-\alpha/2}\frac{S}{\sqrt{n}}\right),$$

   where z_γ denotes the γ-quantile of the $\mathsf{N}(0,1)$ distribution and S is the sampl standard deviation of $\bar{Y}_{\bullet 1}, \ldots, \bar{Y}_{\bullet n}$.

Figure 9.4 shows a typical outcome for the $d = 2$-dimensional case with $K = 5$ classes per dimension, so that the total number of strata is $m = K^d = 25$. The total sample size is $N = 150$, so that each stratum has exactly $n = N/m = 6$ samples.

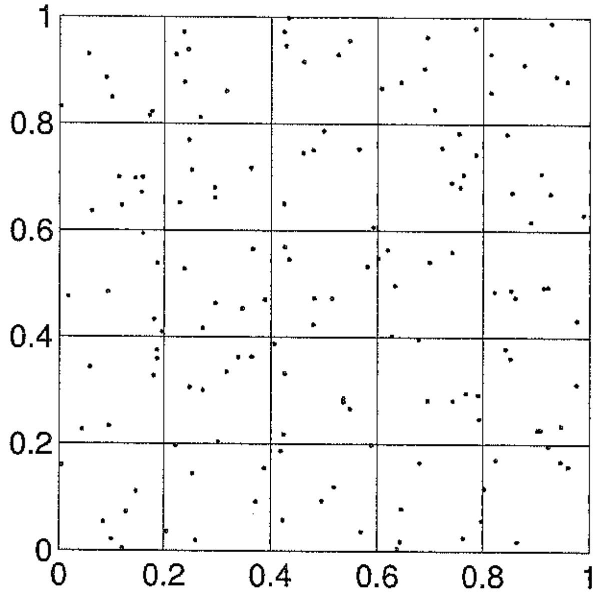


Figure 9.4 Systematic sampling on the unit square. Both dimensions are divided into $K = 5$ classes, giving a total of $m = 25$ strata. Each stratum has $n = 6$ samples for a total sample size of $N = 150$.

■ EXAMPLE 9.5 (Systematic Sampling for the Bridge Network)

We return again to Example 9.1. The MATLAB program below implements Algorithm 9.5 where each of the $d = 5$ dimensions is divided into $K = 4$ classes, giving a total of $m = 4^5 = 1024$ strata. For a total sample size of around $N = 10^4$ (the code below uses $N = 10240$) a typical estimate is $\widehat{\ell}^{(s)} = 0.9301$ with an estimated relative error of 0.13%. This means a tenfold variance reduction in comparison with CMC. Function `h.m` in the code below is the same as in Example 9.1.

```
%stratbridge.m
K = 4;
m = K^5; %number of strata
N = 10^4; %total number of samples
n = ceil(N/m); %number of samples per stratum
est = zeros(n,1);
R=(1:m)';
W=zeros(m,5);
W(:,1)=mod(R,K);
for i=2:5
    W(:,i)=(mod(R,K^i)-mod(R,K^(i-1)))./(K^(i-1));
end
```

```
for j=1:n
    V=(W+rand(m,5))./K;
    est(j)=mean(h(V));
end
mest = mean(est)
percRE = std(est)/sqrt(n)/mest*100
```

9.6 LATIN HYPERCUBE SAMPLING

The main drawback of stratification for high-dimensional estimation problems is that the number of strata grows exponentially in the number of classes. For example, if systematic sampling is applied to the d-dimensional hypercube and each coordinate is divided into K classes, then the number of strata is $m = K^d$, which is only practical for small K and d, say $K = 1, \ldots, 5$ and $d = 1, \ldots, 10$. For higher-dimensional problems a useful remedy is to apply **latin hypercube sampling** instead. The idea is to sample on the d-dimensional hypercube in such a way that only the marginal distributions are stratified. Figure 9.5 provides an illustration. In contrast to the full stratification in Figure 9.4 not all cells have the same number of samples. Instead, both the x and y coordinates are stratified in $K = 5$ classes, with 30 samples per class.

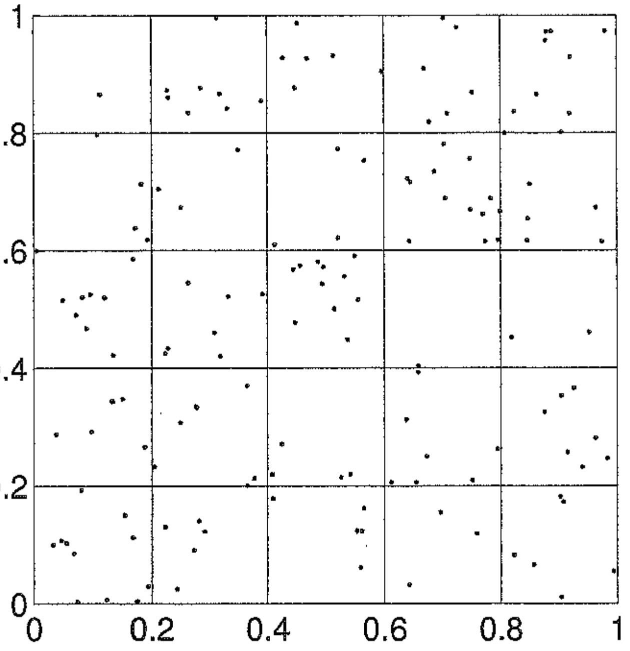


Figure 9.5 Latin hypercube sampling in dimension $d = 2$ with $K = 5$ strata per dimension and $N = 150$ samples. Each one-dimensional stratum has $n = N/K = 30$ samples.

Algorithm 9.6 (Latin Hypercube Sampling) *Starting with $i = 1$, execute the following steps:*

1. *Generate $\mathbf{U}_1, \ldots, \mathbf{U}_K \overset{\text{iid}}{\sim} \mathsf{U}(0,1)^d$.*
2. *Generate K independent uniform permutations, $\mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_K$, of $(1, \ldots, K)$.*
3. *Set*
$$\mathbf{V}_k = \frac{\mathbf{\Pi}_k + \mathbf{1} - \mathbf{U}_k}{K}, \quad k = 1, \ldots, K .$$
Let
$$Y_i = \frac{1}{K} \sum_{k=1}^{K} h(\mathbf{V}_k) .$$
4. *If $i = n$, then go to Step 5; otherwise, set $i = i + 1$ and go to Step 1.*
5. *Estimate ℓ as $\widehat{\ell}^{(h)} = \frac{1}{n}\sum_{i=1}^{n} Y_i$ and determine an approximate $1-\alpha$ confidence interval as*
$$\left(\widehat{\ell}^{(h)} - z_{1-\alpha/2}\frac{S}{\sqrt{n}},\ \widehat{\ell}^{(h)} + z_{1-\alpha/2}\frac{S}{\sqrt{n}}\right),$$
where z_γ denotes the γ-quantile of the $\mathsf{N}(0,1)$ distribution and S is the sample standard deviation of $Y_1, \ldots, Y_n$.

■ EXAMPLE 9.6 (Latin Hypercube Sampling for the Bridge Network)

Consider again Example 9.1. The MATLAB program below implements a latin hypercube sampling scheme with $K = 50$ classes for each of the $d = 5$ dimensions. At each of the $n = 2000$ iterations, 50 points are generated in the five-dimensional hypercube, giving a total of $N = 10^4$ of such points. A typical estimate is $\widehat{\ell}^{(h)} = 0.9287$ with an estimated relative error of 0.16%, which is comparable with the 0.13% of the full stratification in Example 9.5. Function h.m in the code below is the same as in Example 9.1.

```
%lhcsbridge.m
d = 5;
K = 50;
N = 10^4;
n = N/K;
est = zeros(n,1);
for i = 1:n
    U = rand(K,d);
    [x,p] = sort(rand(K,d));
    V = (p + 1 - U)/K;
    est(i) = mean(h(V));
end
mean(est)
percRE = std(est)/sqrt(n)/mean(est)*100
```

9.7 IMPORTANCE SAMPLING

One of the most important variance reduction techniques is **importance san pling**. This technique is especially useful for the estimation of rare-event probab ities (see Chapter 10). The standard setting is the estimation of a quantity

$$\ell = \mathbb{E}_f H(\mathbf{X}) = \int H(\mathbf{x})\, f(\mathbf{x})\, \mathrm{d}\mathbf{x}\,, \qquad (9.1$$

where H is a real-valued function and f the probability density of a random vect $\mathbf{X}$, called the **nominal pdf**. The subscript f is added to the expectation operat to indicate that it is taken with respect to the density f.

Let g be another probability density such that $H\,f$ is **dominated** by g. Th is, $g(\mathbf{x}) = 0 \Rightarrow H(\mathbf{x})\, f(\mathbf{x}) = 0$. Using the density g we can represent ℓ as

$$\ell = \int H(\mathbf{x})\, \frac{f(\mathbf{x})}{g(\mathbf{x})}\, g(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathbb{E}_g H(\mathbf{X})\, \frac{f(\mathbf{X})}{g(\mathbf{X})}\,. \qquad (9.1$$

Consequently, if $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim_{\text{iid}} g$, then

$$\widehat{\ell} = \frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k)\, \frac{f(\mathbf{X}_k)}{g(\mathbf{X}_k)} \qquad (9.1$$

is an unbiased estimator of ℓ. This estimator is called the **importance samplin estimator** and g is called the importance sampling density. The ratio of densitie

$$W(\mathbf{x}) = \frac{f(\mathbf{x})}{g(\mathbf{x})}\,, \qquad (9.1$$

is called the **likelihood ratio** — with a slight abuse of nomenclature, as the like ☞ 664 hood is usually seen in statistics as a function of the parameters (see Section B.

Algorithm 9.7 (Importance Sampling Estimation)

1. *Select an importance sampling density g that dominates Hf.*

2. *Generate $\mathbf{X}_1, \ldots, \mathbf{X}_N \overset{\text{iid}}{\sim} g$ and let $Y_i = H(\mathbf{X}_i) f(\mathbf{X}_i)/g(\mathbf{X}_i)$, $i = 1, \ldots, N$.*

3. *Estimate ℓ via $\widehat{\ell} = \bar{Y}$ and determine an approximate $1-\alpha$ confidence inter as*

$$\left(\widehat{\ell} - z_{1-\alpha/2}\frac{S}{\sqrt{N}},\ \widehat{\ell} + z_{1-\alpha/2}\frac{S}{\sqrt{N}}\right),$$

where z_γ denotes the γ-quantile of the $\mathsf{N}(0,1)$ *distribution and S is the sam standard deviation of $Y_1, \ldots, Y_N$.*

■ **EXAMPLE 9.7 (Importance Sampling for the Bridge Network)**

The expected length of the shortest path in Example 9.1 can be written as (see (9.2))

$$\ell = \mathbb{E}h(\mathbf{U}) = \int h(\mathbf{u})\,\mathrm{d}\mathbf{u}\,,$$

where $\mathbf{U} = (U_1, \ldots, U_5)$ and $U_1, \ldots, U_5 \overset{\text{iid}}{\sim} \mathsf{U}(0,1)$. The nominal pdf is thus $f(\mathbf{u}) = 1, \mathbf{u} \in (0,1)^5$. Suppose the importance sampling pdf is of the form

$$g(\mathbf{u}) = \prod_{i=1}^{5} \nu_i\, u_i^{\nu_i - 1}\,,$$

which means that under g the components of $\mathbf{U}$ are again independent and $U_i \sim \mathsf{Beta}(\nu_i, 1)$ for some $\nu_i > 0$, $i = 1, \ldots, 5$. For the nominal (uniform) distribution we have $\nu_i = 1$, $i = 1, \ldots, 5$. Generating $\mathbf{U}$ under g is easily carried out via the inverse-transform method — see Algorithm 4.18. A good choice of $\{\nu_i\}$ is of ☞ 105 course crucial. The MATLAB program below implements the importance sampling estimation of ℓ using, for example, $(\nu_1, \ldots, \nu_5) = (1.3, 1.1, 1, 1.3, 1.1)$. For a sample size of $N = 10^4$ a typical estimate is $\widehat{\ell} = 0.9295$ with an estimated relative error of 0.24%, which gives about a fourfold reduction in simulation effort compared with CMC estimation, despite the fact that the $\{\nu_i\}$ are all quite close to their nominal value 1.

```
%bridgeIS.m
N = 10^4;
nu0 = [1.3, 1.1, 1, 1.3, 1.1];
nu = repmat(nu0,N,1);
U = rand(N,5).^(1./nu);
W = prod(1./(nu.*U.^(nu - 1)),2);
y = h(U).*W;
est = mean(y)
percRE = std(y)/sqrt(N)/est*100
```

The main difficulty in importance sampling is how to choose the importance sampling distribution. A poor choice of g may seriously compromise the estimate and the confidence intervals. The following sections provide some guidance on choosing a good importance sampling distribution.

9.7.1 Minimum-Variance Density

The optimal choice g^* for the importance sampling density minimizes the variance of $\widehat{\ell}$, and is therefore the solution to the functional minimization program

$$\min_g \mathrm{Var}_g\left(H(\mathbf{X})\,\frac{f(\mathbf{X})}{g(\mathbf{X})}\right)\,. \tag{9.20}$$

It is not difficult to show (see for example [10]) that

$$g^*(\mathbf{x}) = \frac{|H(\mathbf{x})|\, f(\mathbf{x})}{\int |H(\mathbf{x})|\, f(\mathbf{x})\,\mathrm{d}\mathbf{x}}\,. \tag{9.21}$$

In particular, if $H(\mathbf{x}) \geqslant 0$ or $H(\mathbf{x}) \leqslant 0$ then

$$g^*(\mathbf{x}) = \frac{H(\mathbf{x})\, f(\mathbf{x})}{\ell}\,, \qquad (9.22$$

in which case $\mathrm{Var}_{g^*}(\widehat{\ell}) = \mathrm{Var}_{g^*}(H(\mathbf{X})W(\mathbf{X})) = \mathrm{Var}_{g^*}(\ell) = 0$, so that the estimato $\widehat{\ell}$ is *constant* under g^*. An obvious difficulty is that the evaluation of the optima importance sampling density g^* is usually not possible. For example, $g^*(\mathbf{x})$ in (9.22 depends on the unknown quantity ℓ. Nevertheless, a good importance samplin density g should be "close" to the minimum variance density g^*.

One of the main considerations for choosing a good importance sampling pd is that the estimator (9.18) should have finite variance. This is equivalent to th requirement that

$$\mathbb{E}_g H^2(\mathbf{X}) \frac{f^2(\mathbf{X})}{g^2(\mathbf{X})} = \mathbb{E}_f H^2(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})} < \infty\,. \qquad (9.23$$

This suggests that g should not have lighter tails than f, and that, preferably, th likelihood ratio, f/g, should be bounded.

9.7.2 Variance Minimization Method

When the pdf f belongs to some parametric family of distributions, it is ofte convenient to choose the importance sampling distribution from the *same* famil In particular, suppose that $f(\cdot;\boldsymbol{\theta})$ belongs to the family

$$\{f(\cdot;\boldsymbol{\eta}),\ \boldsymbol{\eta} \in \Theta\}\,.$$

Then, the problem of finding an optimal importance sampling density in this clas reduces to the following *parametric* minimization problem

$$\min_{\boldsymbol{\eta}\in\Theta} \mathrm{Var}_{\boldsymbol{\eta}}\left(H(\mathbf{X})\, W(\mathbf{X};\boldsymbol{\theta},\boldsymbol{\eta})\right)\,, \qquad (9.24$$

where $W(\mathbf{X};\boldsymbol{\theta},\boldsymbol{\eta}) = f(\mathbf{X};\boldsymbol{\theta})/f(\mathbf{X};\boldsymbol{\eta})$. We call $\boldsymbol{\theta}$ the **nominal parameter** and the **reference parameter vector** or **tilting vector**. Since under any $f(\cdot;\boldsymbol{\eta})$ th expectation of $H(\mathbf{X})\, W(\mathbf{X};\boldsymbol{\theta},\boldsymbol{\eta})$ is ℓ, the optimal solution of (9.24) coincides wit that of

$$\min_{\boldsymbol{\eta}\in\Theta} V(\boldsymbol{\eta})\,, \qquad (9.25$$

where

$$V(\boldsymbol{\eta}) = \mathbb{E}_{\boldsymbol{\eta}} H^2(\mathbf{X})\, W^2(\mathbf{X};\boldsymbol{\theta},\boldsymbol{\eta}) = \mathbb{E}_{\boldsymbol{\theta}} H^2(\mathbf{X})\, W(\mathbf{X};\boldsymbol{\theta},\boldsymbol{\eta})\,. \qquad (9.26$$

We call either of the equivalent problems (9.24) and (9.25) the **variance mini mization** (VM) problem; and we call the parameter vector ${}_*\boldsymbol{\eta}$ that minimizes th programs (9.24) and (9.25) the **VM-optimal reference parameter vector**. Th VM problem can be viewed as a stochastic optimization problem, and can be ap proximately solved via Monte Carlo simulation by considering the sample averag
☞ 446 version of (9.25) and (9.26):

$$\min_{\boldsymbol{\eta}\in\Theta} \widehat{V}(\boldsymbol{\eta})\,, \qquad (9.27$$

where

$$\widehat{V}(\boldsymbol{\eta}) = \frac{1}{N}\sum_{k=1}^{N} H^2(\mathbf{X}_k)\, W(\mathbf{X}_k;\boldsymbol{\theta},\boldsymbol{\eta}) , \tag{9.28}$$

and $\mathbf{X}_1,\ldots,\mathbf{X}_N \sim_{\text{iid}} f(\cdot;\boldsymbol{\theta})$. This problem can be solved via standard numerical optimization techniques. This gives the following modification of Algorithm 9.7.

Algorithm 9.8 (Variance Minimization Method)

1. *Select a parameterized family of importance sampling densities* $\{f(\cdot;\boldsymbol{\eta})\}$.
2. *Generate a pilot sample* $\mathbf{X}_1,\ldots,\mathbf{X}_N \overset{\text{iid}}{\sim} f(\cdot;\boldsymbol{\theta})$*, and determine the solution* ${}_*\widehat{\boldsymbol{\eta}}$ *to the variance minimization problem* (9.27).
3. *Generate* $\mathbf{X}_1,\ldots,\mathbf{X}_{N_1} \overset{\text{iid}}{\sim} f(\cdot;\,{}_*\widehat{\boldsymbol{\eta}})$ *and let* $Y_i = H(\mathbf{X}_i) f(\mathbf{X}_i;\boldsymbol{\theta})/f(\mathbf{X}_i;\,{}_*\widehat{\boldsymbol{\eta}}), i = 1,\ldots,N_1$.
4. *Estimate* ℓ *via* $\widehat{\ell} = \bar{Y}$ *and determine an approximate* $1-\alpha$ *confidence interval as*
$$\left(\widehat{\ell} - z_{1-\alpha/2}\frac{S}{\sqrt{N_1}},\ \widehat{\ell} + z_{1-\alpha/2}\frac{S}{\sqrt{N_1}}\right),$$
where z_γ *denotes the* γ*-quantile of the* $\mathsf{N}(0,1)$ *distribution and* S *is the sample standard deviation of* $Y_1,\ldots,Y_{N_1}$.

■ **EXAMPLE 9.8 (Variance Minimization for the Bridge Network)**

Consider the importance sampling approach for the bridge network in Example 9.7. There, the importance sampling distribution is the joint distribution of independent $\mathsf{Beta}(\nu_i, 1)$ random variables, for $i = 1,\ldots,5$. Hence, the reference parameter is $\boldsymbol{\nu} = (\nu_1,\ldots,\nu_5)$.

The following MATLAB program determines the optimal reference parameter vector ${}_*\widehat{\boldsymbol{\nu}}$ via the VM method using a pilot run of size $N = 10^3$ and the standard MATLAB minimization routine `fminsearch`. A typical value for ${}_*\widehat{\boldsymbol{\nu}}$ is $(1.262, 1.083, 1.016, 1.238, 1.067)$, which is similar to the one used in Example 9.7; the relative error is thus around 0.24%.

```
%vmceopt.m
N = 10^3;
U = rand(N,5);
[nu0,minv] =fminsearch(@(nu)f_var(nu,U,N),ones(1,5))
N1 = 10^4;
nu = repmat(nu0,N1,1);
U = rand(N1,5).^(1./nu);
w = prod(1./(nu.*U.^(nu - 1)),2);
y = h(U).*w;
est = mean(y)
percRE = std(y)/sqrt(N1)/est*100
```

```
function out = f_var(nu,U,N)
nu1 = repmat(nu,N,1);
W = prod(1./(nu1.*U.^(nu1 - 1)),2);
y = H(U);
out = W'*y.^2;
```

9.7.3 Cross-Entropy Method

An alternative approach to the VM method for choosing an "optimal" importance sampling distribution is based on the Kullback–Leibler cross-entropy distance, or simply **cross-entropy** (CE) distance. The CE distance between two continuous pdfs g and h is given by

$$\begin{aligned}\mathcal{D}(g,h) &= \mathbb{E}_g \ln \frac{g(\mathbf{X})}{h(\mathbf{X})} = \int g(\mathbf{x}) \ln \frac{g(\mathbf{x})}{h(\mathbf{x})}\,\mathrm{d}\mathbf{x} \\ &= \int g(\mathbf{x}) \ln g(\mathbf{x})\,\mathrm{d}\mathbf{x} - \int g(\mathbf{x}) \ln h(\mathbf{x})\,\mathrm{d}\mathbf{x}\,.\end{aligned} \tag{9.29}$$

☞ 614

For discrete pdfs replace the integrals with the corresponding sums. Observe that, by Jensen's inequality, $\mathcal{D}(g,h) \geqslant 0$, with equality if and only if $g = h$. The CE distance is sometimes called the Kullback–Leibler *divergence*, because it is not symmetric, that is, $\mathcal{D}(g,h) \neq \mathcal{D}(h,g)$ for $g \not\equiv h$.

The idea of the CE method is to choose the importance sampling density, h say, such that the CE distance between the optimal importance sampling density g^* in (9.21) and h, is minimal. We call this the **CE-optimal pdf**. This pdf solves the *functional* optimization program $\min_h \mathcal{D}\,(g^*, h)$. If we optimize over all densities h, then it is immediate that the CE-optimal pdf coincides with the VM-optimal pdf g^*.

As with the VM approach in (9.24) and (9.25), we shall restrict ourselves to a parametric family of densities $\{f(\cdot;\boldsymbol{\eta}), \boldsymbol{\eta} \in \Theta\}$ that contains the nominal density $f(\cdot;\boldsymbol{\theta})$. Moreover, without any loss of generality, we only consider positive functions H. The CE method now aims to solve the *parametric* optimization problem

$$\min_{\boldsymbol{\eta}\in\Theta} \mathcal{D}\left(g^*, f(\cdot;\boldsymbol{\eta})\right)\,. \tag{9.30}$$

The optimal solution coincides with that of

$$\max_{\boldsymbol{\eta}\in\Theta} D(\boldsymbol{\eta})\,, \tag{9.31}$$

where

$$D(\boldsymbol{\eta}) = \mathbb{E}_{\boldsymbol{\theta}} H(\mathbf{X})\, \ln f(\mathbf{X};\boldsymbol{\eta})\,. \tag{9.32}$$

Similar to the VM program (9.25), we call either of the equivalent programs (9.30) and (9.31) the **CE program**; and we call the parameter vector $\boldsymbol{\eta}^*$ that minimizes the program (9.30) and (9.31) the **CE-optimal reference parameter**.

☞ 446

Similar to (9.27) we can estimate $\boldsymbol{\eta}^*$ via the stochastic counterpart method as the solution of the stochastic program

$$\max_{\boldsymbol{\eta}} \widehat{D}(\boldsymbol{\eta}) = \max_{\boldsymbol{\eta}} \frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k) \ln f(\mathbf{X}_k; \boldsymbol{\eta}) , \tag{9.33}$$

where $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim_{\text{iid}} f(\cdot; \boldsymbol{\theta})$.

In typical applications the function $\widehat{D}$ in (9.33) is convex and differentiable with respect to $\boldsymbol{\eta}$ (see [19]). In such cases the solution of (9.33) may be obtained by solving (with respect to $\boldsymbol{\eta}$) the following system of equations:

$$\frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k) \nabla \ln f(\mathbf{X}_k; \boldsymbol{\eta}) = \mathbf{0} , \tag{9.34}$$

where the gradient is with respect to $\boldsymbol{\eta}$. Various numerical and theoretical studies [17] have shown that the solutions to the VM and CE programs are qualitatively similar. The main advantage of the CE approach over the VM approach is that the solution to (9.33) (or (9.34)) can often be found *analytically*, as specified in the following theorem. A proof can be found in [17, Pages 69–70].

Theorem 9.7.1 (Exponential Families) *If the importance sampling density is of the form*

$$f(\mathbf{x}; \boldsymbol{\eta}) = \prod_{i=1}^{n} f_i(x_i; \eta_i) ,$$

where each $\{f_i(x_i; \eta_i), \eta_i \in \Theta_i\}$ forms a 1-parameter exponential family parameterized by the mean, then the solution to the CE program (9.33) *is $\widehat{\boldsymbol{\eta}}^* = (\widehat{\eta}_1^*, \ldots, \widehat{\eta}_n^*)$, with* ☞ 701

$$\widehat{\eta}_i^* = \frac{\sum_{k=1}^{N} H(\mathbf{X}_k)\, X_{ki}}{\sum_{k=1}^{N} H(\mathbf{X}_k)} , \quad i = 1, \ldots, n , \tag{9.35}$$

where X_{ki} is the i-th coordinate of $\mathbf{X}_k$.

For rare-event simulation the random variable $H(\mathbf{X})$ often takes the form of an indicator $\mathrm{I}_{\{S(\mathbf{X}) \geqslant \gamma\}}$. If the event $\{S(\mathbf{X}) \geqslant \gamma\}$ is rare under $f(\cdot; \boldsymbol{\theta})$, then with high probability the numerator and denominator in (9.35) are both zero, so that the CE-optimal parameter cannot be estimated in this way. Section 10.5 discusses how this can be remedied by using a multilevel approach or by sampling directly from the zero-variance importance sampling pdf g^*. ☞ 404

■ EXAMPLE 9.9 (CE Method for the Bridge Network)

In Example 9.8 the VM-optimal reference parameter is obtained by numerical minimization. We can use the CE method instead by applying (9.35) after suitable reparameterization. Note that for each i, $\mathsf{Beta}(\nu_i, 1)$ forms an exponential family, and that the corresponding expectation is $\eta_i = \nu_i/(1 + \nu_i)$. It follows that the assignment $\nu_i = \eta_i/(1 - \eta_i)$ reparameterizes the family in terms of the mean η_i.

The first four lines of the following MATLAB program implement the CE method for estimating the CE-optimal reference parameter. A typical outcome is $\widehat{\boldsymbol{\eta}} = (0.560, 0.529, 0.500, 0.571, 0.518)$, so that $\widehat{\boldsymbol{\nu}} = (1.272, 1.122, 1.000, 1.329, 1.075)$, which gives comparable results to the VM-optimal parameter vector. The corresponding relative error is estimated as 0.25%.

```
%bridgeCE.m
N = 10^3;
U = rand(N,5);
y = repmat(h(U),1,5);
v = sum(y.*U)./sum(y)
N1 = 10^4;
nu = repmat(v./(1-v),N1,1);
U = rand(N1,5).^(1./nu);
w = prod(1./(nu.*U.^(nu - 1)),2);
y = h(U).*w;
est = mean(y)
percRE = std(y)/sqrt(N1)/est*100
```

9.7.4 Weighted Importance Sampling

Algorithm 9.7 can be modified slightly by allowing the likelihood ratio (9.19) to be known *up to a constant*; that is, $W(\mathbf{X}) = f(\mathbf{X})/g(\mathbf{X}) = c\,w(\mathbf{X})$ for some known function $w(\cdot)$, but possibly unknown constant c. Since $\mathbb{E}_g W(\mathbf{X}) = 1$, we can write $\ell = \mathbb{E}_g H(\mathbf{X})\, W(\mathbf{X})$ as

$$\ell = \frac{\mathbb{E}_g H(\mathbf{X})\, W(\mathbf{X})}{\mathbb{E}_g W(\mathbf{X})} \; .$$

This suggests as an alternative to the standard importance sampling estimator (9.18) the following **weighted sample estimator**:

$$\widehat{\ell}_w = \frac{\sum_{k=1}^{N} H(\mathbf{X}_k)\, w_k}{\sum_{k=1}^{N} w_k} \; . \tag{9.36}$$

Here the $\{w_k\}$, with $w_k = w(\mathbf{X}_k)$, are interpreted as *weights* of the random sample $\{\mathbf{X}_k\}$, and the population $\{(\mathbf{X}_k, w_k)\}$ is called a **weighted sample** from $g(\mathbf{x})$. Since the estimator is a ratio estimator (see Example 8.4), the weighted sample estimator (9.36) introduces some bias, which tends to 0 as N increases. Loosely speaking, we may view the weighted sample $\{(\mathbf{X}_k, w_k)\}$ as a representation of $f(\mathbf{x})$, in the sense that $\ell = \mathbb{E}_f H(\mathbf{X}) \approx \widehat{\ell}_w$ for any function H.

☞ 308

■ **EXAMPLE 9.10 (Weighted Sample Estimator for the Bridge Network)**

For the bridge example it is tempting to use the zero-variance pdf $g^*(\mathbf{u}) = \frac{1}{\ell}\, h(\mathbf{u})$ as the importance sampling pdf, in conjunction with (9.36). Note that $h(\mathbf{u}) \leqslant 2$ for all $\mathbf{u}$. Sampling from g^* can therefore be done via acceptance–rejection: sample $\mathbf{U} \sim \mathsf{U}(0,1)^5$ and $V \sim \mathsf{U}(0,2)$ independently, and accept $\mathbf{U}$ if $V < h(\mathbf{U})$. Since $w(\mathbf{u}) = 1/h(\mathbf{u})$, the weighted sample estimator becomes

$$\widehat{\ell}_w = \frac{1}{\frac{1}{N}\sum_{k=1}^{N} 1/h(\mathbf{U}_k)} \, ,$$

where $\mathbf{U}_1, \ldots, \mathbf{U}_N \overset{\text{iid}}{\sim} g^*$. By the delta method, $\sqrt{N}(\widehat{\ell}_w - \ell)$ converges in distribution ☞ 308 to a $\mathsf{N}(0, \sigma^2 \ell^4)$ distribution, where σ^2 is the variance of $1/h(\mathbf{U})$, which can be readily estimated from the simulation. The MATLAB program below implements the weighted sample estimator. Although the program produces unbiased estimates, the relative error is around 0.65%, which is *worse* than the one for CMC.

```
%wsbridge.m
clear all
N = 10^4;w  = zeros(N,1);
for k=1:N
    cont = true;
    while cont
        R = rand(1,5);v = rand*2;y = h(R);
        if v < y
            w(k) = 1/y;cont=false;
        end
    end
end
est = 1/mean(w)
percRE = std(w)*est/sqrt(N)*100
```

9.7.5 Sequential Importance Sampling

Sequential importance sampling (SIS), also called **dynamic importance sampling**, is simply importance sampling carried out in a sequential manner. To explain the SIS procedure, consider the expected performance $\ell = \mathbb{E}_f H(\mathbf{X})$ and its importance sampling estimator

$$\widehat{\ell} = \frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k) \, \frac{f(\mathbf{X}_k)}{g(\mathbf{X}_k)} , \quad \mathbf{X}_1, \ldots, \mathbf{X}_N \overset{\text{iid}}{\sim} g , \tag{9.37}$$

where g is the importance sampling pdf. Suppose that (a) $\mathbf{X}$ can be written as a vector $\mathbf{X} = (X_1, \ldots, X_n)$, where each of the X_i may be multidimensional; and (b) it is easy to sample from $g(\mathbf{x})$ sequentially. Specifically, suppose that $g(\mathbf{x})$ is of the form

$$g(\mathbf{x}) = g_1(x_1)\, g_2(x_2 \,|\, x_1) \cdots g_n(x_n \,|\, x_1, \ldots, x_{n-1}) , \tag{9.38}$$

where it is easy to generate X_1 from the density $g_1(x_1)$, and conditional on $X_1 = x_1$ the second component from the density $g_2(x_2 \,|\, x_1)$, and so on, until one obtains a single random vector $\mathbf{X}$ from $g(\mathbf{x})$. Repeating this independently N times, one obtains an iid sample $\mathbf{X}_1, \ldots, \mathbf{X}_N$ from $g(\mathbf{x})$ and estimates ℓ according to (9.37).

To further simplify the notation we abbreviate $(x_1, \ldots, x_t)$ to $\mathbf{x}_{1:t}$ for all t. In particular, $\mathbf{x}_{1:n} = \mathbf{x}$. Typically, t can be viewed as a (discrete) time parameter and $\mathbf{x}_{1:t}$ as a path or trajectory. By the product rule of probability (A.21), the target ☞ 616 pdf $f(\mathbf{x})$ can also be written sequentially as

$$f(\mathbf{x}) = f(x_1)\, f(x_2 \,|\, x_1) \cdots f(x_n \,|\, \mathbf{x}_{1:n-1}) , \tag{9.39}$$

☞ 672 where we use a Bayesian notational convention (see Section B.3) for notational convenience. From (9.38) and (9.39) it follows that we can write the likelihood ratio in product form as

$$W(\mathbf{x}) = \frac{f(x_1)\, f(x_2 \,|\, x_1) \cdots f(x_n \,|\, \mathbf{x}_{1:n-1})}{g_1(x_1)\, g_2(x_2 \,|\, x_1) \cdots g_n(x_n \,|\, \mathbf{x}_{1:n-1})} \,. \tag{9.40}$$

If $W_t(\mathbf{x}_{1:t})$ denotes the likelihood ratio up to time t, we can write it recursively as

$$W_t(\mathbf{x}_{1:t}) = u_t\, W_{t-1}(\mathbf{x}_{1:t-1}), \quad t = 1, \ldots, n \,, \tag{9.41}$$

with initial weight $W_0(\mathbf{x}_{1:0}) = 1$ and **incremental weights** $u_1 = f(x_1)/g_1(x_1)$ and

$$u_t = \frac{f(x_t \,|\, \mathbf{x}_{1:t-1})}{g_t(x_t \,|\, \mathbf{x}_{1:t-1})} = \frac{f(\mathbf{x}_{1:t})}{f(\mathbf{x}_{1:t-1})\, g_t(x_t \,|\, \mathbf{x}_{1:t-1})} \,, \quad t = 2, \ldots, n \,. \tag{9.42}$$

In order to update the likelihood recursively as in (9.42) one needs to know the marginal pdf $f(\mathbf{x}_{1:t})$ for each t. This may not be simple when f does not have a Markov structure, as it requires integrating $f(\mathbf{x})$ over all $x_{t+1}, \ldots, x_n$. Instead, one can introduce a sequence of *auxiliary* pdfs $f_1, f_2, \ldots, f_n$ that are easily evaluated, and such that each $f_t(\mathbf{x}_{1:t})$ is a good approximation to $f(\mathbf{x}_{1:t})$. The terminating pdf f_n must be equal to the original f. Since

$$f(\mathbf{x}) = \frac{f_1(x_1)}{1} \frac{f_2(\mathbf{x}_{1:2})}{f_1(x_1)} \cdots \frac{f_n(\mathbf{x}_{1:n})}{f_{n-1}(\mathbf{x}_{1:n-1})} \,, \tag{9.43}$$

as a generalization of (9.42) we have the incremental updating weight

$$u_t = \frac{f_t(\mathbf{x}_{1:t})}{f_{t-1}(\mathbf{x}_{1:t-1})\, g_t(x_t \,|\, \mathbf{x}_{1:t-1})} \,, \tag{9.44}$$

for $t = 1, \ldots, n$, where we put $f_0(\mathbf{x}_{1:0}) = 1$. Note that the incremental weights u_t only need to be defined *up to a constant*, say c_t, for each t. In this case the likelihood ratio $W(\mathbf{x})$ is known up to a constant as well, say $W(\mathbf{x}) = C\, w(\mathbf{x})$, where $1/C = \mathbb{E}_g w(\mathbf{X})$ can be estimated via the corresponding sample mean. In other words, when the normalization constant is unknown, one can still estimate ℓ using the weighted sample estimator (9.36) rather than the importance sampling estimator (9.18).

Summarizing, the SIS method can be written as follows.

Algorithm 9.9 (Sequential Importance Sampling)

1. *For each $t = 1, \ldots, n$, sample X_t from $g_t(x_t \,|\, \mathbf{x}_{1:t-1})$.*

2. *Compute $w_t = u_t\, w_{t-1}$, where $w_0 = 1$ and*

$$u_t = \frac{f_t(\mathbf{X}_{1:t})}{f_{t-1}(\mathbf{X}_{1:t-1})\, g_t(X_t \,|\, \mathbf{X}_{1:t-1})}, \quad t = 1, \ldots, n \,. \tag{9.45}$$

3. *Repeat the steps above N times and estimate ℓ via $\widehat{\ell}$ in* (9.18) *or $\widehat{\ell}_w$ in* (9.36).

Applications of sequential importance sampling frequently involve random
results concerning stopping times are

Theorem 9.7.2 (Importance Sampling With a Stopping Time) *Let τ be a stopping time with respect to the stochastic process $\{X_t, t = 1, 2, \ldots\}$. Let $\mathbb{P}$ and $\widetilde{\mathbb{P}}$ be two measures under which $\mathbf{X}_{1:t} = (X_1, \ldots, X_t)$ has pdf $f_t(\mathbf{x}_{1:t})$ and $g_t(\mathbf{x}_{1:t})$, respectively, for $t = 1, 2, \ldots$. Then, for each sequence of real-valued functions H_t of $\mathbf{x}_{1:t}$, $t = 1, 2, \ldots$,*

$$\mathbb{E} \sum_{t=1}^{\tau} H_t(\mathbf{X}_{1:t}) = \widetilde{\mathbb{E}} \sum_{t=1}^{\tau} H_t(\mathbf{X}_{1:t})\, W_t \,, \tag{9.46}$$

and

$$\mathbb{E} H_\tau(\mathbf{X}_{1:\tau}) = \widetilde{\mathbb{E}} H_\tau(\mathbf{X}_{1:\tau})\, W_\tau \,, \tag{9.47}$$

where $W_t = f_t(\mathbf{X}_{1:t})/g_t(\mathbf{X}_{1:t})$ is the likelihood ratio of $\mathbf{X}_{1:t}$.

■ EXAMPLE 9.11 (Random Walk on the Integers)

Consider a random walk process $\{S_t, t = 0, 1, \ldots\}$ on the integers: $S_t = S_{t-1} + X_t$, where the $\{X_t\}$ are independent $\mathbb{P}(X_t = 1) = p$ and $\mathbb{P}(X_t = -1) = q = 1 - p$ for all $t = 1, 2, \ldots$. Suppose that $p < q$, so that the walk has a drift toward $-\infty$. Our goal is to estimate the rare-event probability ℓ of reaching state K before state 0, starting from state $0 < k \ll K$, where K is a large number. Let $\widetilde{\mathbb{P}}$ be the probability measure under which the $\{X_t\}$ are again independent, but now with $\widetilde{\mathbb{P}}(X_t = 1) = \widetilde{p}$ and $\widetilde{\mathbb{P}}(X_t = -1) = \widetilde{q} = 1 - \widetilde{p}$ for all $t = 1, 2, \ldots$. Define τ as the first time that either 0 or K is reached. As τ is a stopping time for $\{S_t\}$ we have by (9.47) that

$$\ell = \mathbb{E}\, \mathrm{I}_{\{S_\tau = K\}} = \widetilde{\mathbb{E}}\, \mathrm{I}_{\{S_\tau = K\}} W_\tau \,,$$

where $W_t, t = 1, 2, \ldots$ can be computed sequentially as $W_t = W_{t-1}\, u_t$ with

$$u_t = \begin{cases} p/\widetilde{p} & \text{if } x_t = 1 \,, \\ q/\widetilde{q} & \text{if } x_t = -1 \,, \end{cases}$$

where $W_0 = 1$. Consider the exponential family of pdfs $\{f(x; \theta),\ \theta \in \mathbb{R}\}$ defined by

$$f(x; \theta) = \mathrm{e}^{\theta x - \zeta(\theta)} f_0(x), \quad x \in \{-1, 1\} \,,$$

where $f_0(1) = p$ and $f_0(-1) = q$ (corresponding to the nominal pdf of X_t) and $\zeta(\theta) = \ln(p\mathrm{e}^{\theta} + q\mathrm{e}^{-\theta})$. Note that $\widetilde{p}$ can be related to θ via $\widetilde{p} = p\mathrm{e}^{\theta}/(p\mathrm{e}^{\theta} + q\mathrm{e}^{-\theta})$. The family can be reparameterized by the mean $v = \zeta'(\theta) = (p\mathrm{e}^{\theta} - q\mathrm{e}^{-\theta})/(p\mathrm{e}^{\theta} + q\mathrm{e}^{-\theta})$. The CE-optimal parameter v^* for estimating ℓ can be derived similarly to Theorem 9.7.1 and is given by (see [4]):

$$v^* = \frac{\mathbb{E}\, \mathrm{I}_{\{S_\tau = K\}} \sum_{i=1}^{\tau} X_i}{\mathbb{E} \tau\, \mathrm{I}_{\{S_\tau = K\}}} = \frac{(K - k)\, \mathbb{P}(S_\tau = K)}{\mathbb{E} \tau\, \mathrm{I}_{\{S_\tau = K\}}} = \frac{K - k}{\mathbb{E}[\tau \,|\, S_\tau = K]} \,.$$

Stern [21] shows that

$$\mathbb{E}\,[\tau \,|\, S_\tau = K] = \frac{1}{(p - q)(1 - r^k)} \left[(K - k)(r^k + 1) + 2K \left(\frac{r^k - r^K}{r^K - 1} \right) \right],$$

where $r = q/p$. Thus, the CE-optimal tilting parameter is

$$v^* = \frac{(K-k)(p-q)(1-r^k)}{(K-k)(r^k+1) + 2K\left(\frac{r^k - r^K}{r^K - 1}\right)} \ .$$

The likelihood ratio of $\mathbf{X}_{1:t} = (X_1, \ldots, X_t)$ is given by

$$W_t = \prod_{i=1}^{t} \frac{f_0(x_i)}{f(x_i;\theta)} = \mathrm{e}^{-\theta \sum_{i=1}^{t} x_i + t\,\zeta(\theta)} \ ,$$

where θ is related to v via $\theta = \frac{1}{2}\ln\left((1+v)q/((1-v)p)\right)$. It follows that under CE-optimal tilting,

$$\mathrm{I}_{\{S_\tau = K\}}\, W_\tau = \mathrm{I}_{\{S_\tau = K\}} \mathrm{e}^{-\theta^*(K-k)+\tau\zeta(\theta^*)} \ .$$

In the MATLAB code below the CE-optimal importance sampling procedure for estimating ℓ is carried out for the case $k = 10$, $K = 30$, and $p = 0.3$. The actual probability is given by

$$\ell = \frac{r^k - 1}{r^K - 1} \approx 4.3689140 \times 10^{-8} \ .$$

A typical estimate using $N = 10^4$ samples is 4.3685×10^{-8}, with an estimated relative error of 1.7×10^{-4}. Through experimentation we observed that the relative error is severely underestimated if N is too small, for example if $N = 1000$ for this case.

```
%gamble_CE_A.m
N = 10^4; %Run Size
results = zeros(N,1);
k = 10; K = 30; %Initial value and absorbing barrier
p = .3; q = 1 - p; r = q/p; %Actual probabilities

%Tilt the distribution using CE
v = ((K - k)*(p - q)*(1 - r^k)) / ((K-k)*(r^k + 1) ...
     + 2 * K * ((r^k - r^K) / (r^K - 1)))
theta = .5*(log((1+v)*q)-log((1-v)*p));
p_tilde = (p * exp(theta)) / (p * exp(theta) + q * exp(-theta));
q_tilde = 1 - p_tilde;

for i = 1:N
    t = 0;
    sum = k;
    while (sum ~= K) && (sum ~= 0)
        t = t+1;
        U = rand;
        sum = sum + (2*(U < p_tilde) - 1);
    end
    results(i) = exp(-theta * (K - k) + t*(log(p*exp(theta) ...
                    + q*exp(-theta))))*(sum == K);
end

ell = (r^k - 1) / (r^K - 1) %Actual Probability
ell_hat = mean(results) %Estimated Probability
RE = std(results) / sqrt(N) / ell_hat %Estimated Relative Error
```

9.7.6 Response Surface Estimation via Importance Sampling

Let the performance measure of a simulation be of the form

$$\ell(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}} H(\mathbf{X}) \, ,$$

where $\mathbf{X} \sim f(\mathbf{x}; \boldsymbol{\theta})$ depends on a parameter $\boldsymbol{\theta} \in \Theta$. Importance sampling makes it possible to gain information on a subset of the **response surface** $\{\ell(\boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta\}$ from a single simulation run [2]. The idea is to write $\ell(\boldsymbol{\theta})$ as

$$\ell(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}_0} H(\mathbf{X}) \, W(\mathbf{X}; \boldsymbol{\theta}, \boldsymbol{\theta}_0) \, , \tag{9.48}$$

where $\boldsymbol{\theta}_0 \in \Theta$ is such that $f(\mathbf{x}; \boldsymbol{\theta}_0) = 0$ dominates $f(\mathbf{x}; \boldsymbol{\theta}) H(\mathbf{x})$. As usual, $W(\mathbf{X}; \boldsymbol{\theta}, \boldsymbol{\theta}_0) = f(\mathbf{X}; \boldsymbol{\theta}) / f(\mathbf{X}; \boldsymbol{\theta}_0)$ is the likelihood ratio. The corresponding estimator is

$$\widehat{\ell}(\boldsymbol{\theta}; \boldsymbol{\theta}_0) = \frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k) \, W(\mathbf{X}_k; \boldsymbol{\theta}, \boldsymbol{\theta}_0) \, , \tag{9.49}$$

where $\mathbf{X}_1, \ldots, \mathbf{X}_N \overset{\text{iid}}{\sim} f(\mathbf{x}; \boldsymbol{\theta}_0)$. The variance of the estimator $\widehat{\ell}(\boldsymbol{\theta}; \boldsymbol{\theta}_0)$ under the importance sampling density $f(\mathbf{x}; \boldsymbol{\theta}_0)$ can be estimated by the sample variance of $\{H(\mathbf{X}_k) W(\mathbf{X}_k; \boldsymbol{\theta}, \boldsymbol{\theta}_0)\}$. The procedure is summarized in the following algorithm.

Algorithm 9.10 (Response Surface Estimation)

1. *Generate* $\mathbf{X}_1, \ldots, \mathbf{X}_N \overset{\text{iid}}{\sim} f(\mathbf{x}; \boldsymbol{\theta}_0)$.

2. *Estimate* $\ell(\boldsymbol{\theta})$ *via* (9.49) *and determine an approximate* $1 - \alpha$ *confidence interval as*

$$\left(\widehat{\ell} - z_{1-\alpha/2} \frac{S}{\sqrt{N}}, \; \widehat{\ell} + z_{1-\alpha/2} \frac{S}{\sqrt{N}} \right),$$

where z_γ *denotes the* γ*-quantile of the* $\mathsf{N}(0,1)$ *distribution and* S *is the sample standard deviation of* $\{H(\mathbf{X}_k) W(\mathbf{X}_k; \boldsymbol{\theta}, \boldsymbol{\theta}_0)\}$.

Two *advantages* of the above procedure are:

1. Only a single simulation run (under $\boldsymbol{\theta}_0$) is needed to estimate the performance for many different values of $\boldsymbol{\theta}$.

2. The estimated response surface $\widehat{\ell}(\boldsymbol{\theta}; \boldsymbol{\theta}_0)$ as a function of $\boldsymbol{\theta}$ is typically piecewise differentiable, allowing easy estimation of the gradient of $\ell(\boldsymbol{\theta})$ through differentiation of $\widehat{\ell}(\boldsymbol{\theta}; \boldsymbol{\theta}_0)$. This is the idea behind the score function method for gradient estimation; see Section 11.4. ☞ 428

The main *disadvantage* is that although $\widehat{\ell}(\boldsymbol{\theta};\boldsymbol{\theta}_0)$ is an unbiased estimator of $\ell(\boldsymbol{\theta})$, its variance can be very large (or even infinite) depending on $(\boldsymbol{\theta},\boldsymbol{\theta}_0)$. This is typically the case when under $\boldsymbol{\theta}_0$ the distribution has thinner tails than under $\boldsymbol{\theta}$, which leads to a blowing up of the likelihood ratio. In such cases the estimator typically *underestimates* the true value and the standard error, so that the confidence intervals are unreliable (too small). Therefore, one should not expect to be able to reliably estimate the *whole* response surface from a sample, but only
☞ 692 a subset thereof, sometimes called the *trust region*; see also Section C.2.2.6 and
☞ 430 Section 11.4.1. A discussion on the dangers of importance sampling may be found, for example, in [18, Pages 209–211].

■ EXAMPLE 9.12 (Response Surface for the Bridge Network)

We return to Example 9.1. Let the lengths $X_1, \ldots, X_5$ of the links be independent and uniformly distributed on $(0,\theta), (0,2), (0,3), (0,1)$, and $(0,2)$, respectively. Hence, the only change in the setting of Example 9.1 is that the first component has a $\mathsf{U}(0,\theta)$ distribution, rather than a $\mathsf{U}(0,1)$ distribution. Denote the expected length of the shortest path by $\ell(\theta)$. Suppose that N iid copies of $\mathbf{X} = (X_1, \ldots, X_5)$ are available for the case where $\theta = \theta_0 = 3$. Then $\ell(\theta)$ can be estimated via

$$\widehat{\ell}(\theta;\theta_0) = \frac{1}{N}\sum_{k=1}^{N} H(\mathbf{X}_k)\frac{\mathrm{I}_{\{0<X_{k1}<\theta\}}/\theta}{\mathrm{I}_{\{0<X_{k1}<\theta_0\}}/\theta_0}\,.$$

Note that for $\theta > \theta_0$ the importance sampling pdf $f(\mathbf{x};\theta_0)$ does not dominate $H(\mathbf{x})f(\mathbf{x};\theta)$, so $\ell(\theta)$ can only be estimated via importance sampling for $\theta < \theta_0$. Figure 9.6 depicts a typical estimate for the response curve for the case $\theta_0 = 3$ using $N = 10^4$ samples.

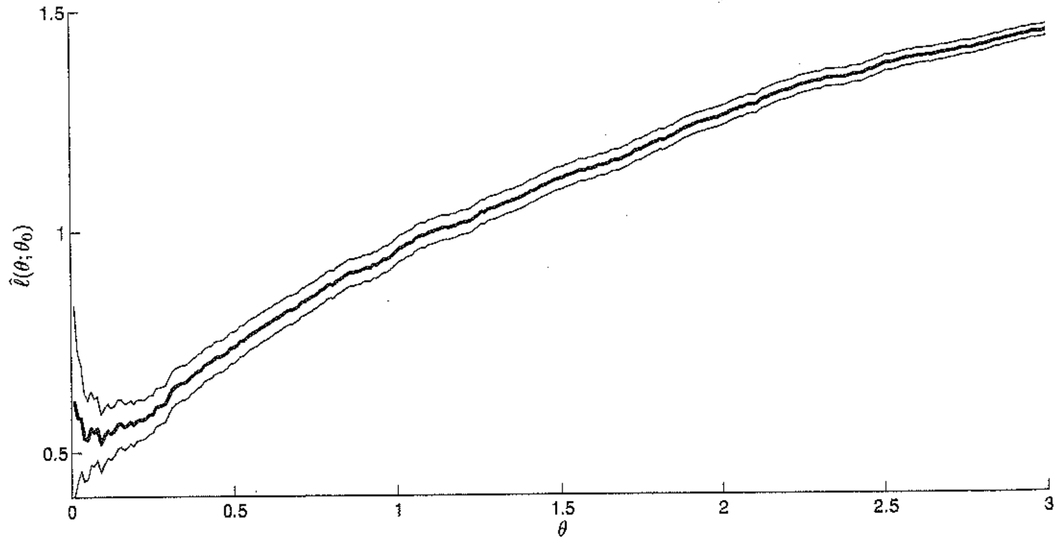


Figure 9.6 Response surface estimates with 95% confidence bounds for the uniform case.

The simulation is implemented via the following MATLAB program.

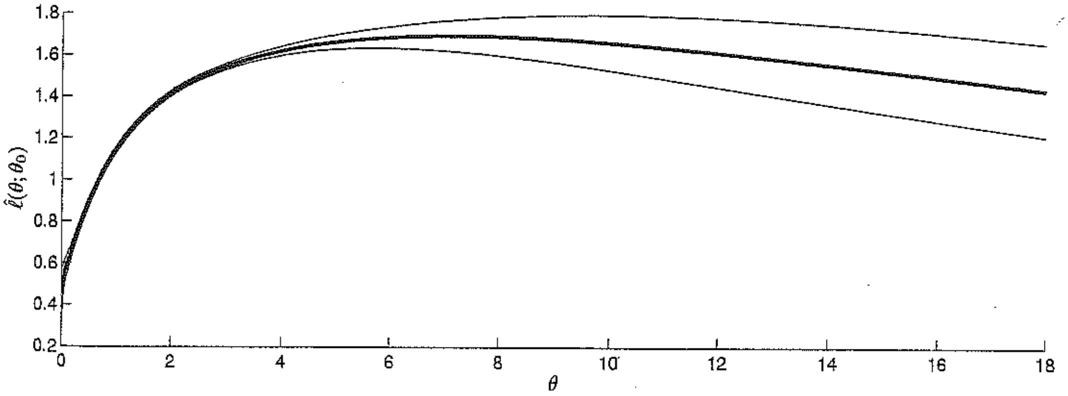
```
%responsesurfis.m
N = 10000; theta0 = 3;
a = [theta0,2,3,1,2]; u = rand(N,5);
X = u.*repmat(a,N,1); W = zeros(N,1);
y = H1(X); theta = 0:0.01:theta0;
num = numel(theta);
ell = zeros(1,num); ellL = zeros(1,num);
ellU = zeros(1,num); stell = zeros(1,num);
for i=1:num
    th = theta(i);
    W = theta0/th*(X(:,1)< th);
    ell(i) = mean(H1(X).*W);
    stell(i) = std(H1(X).*W);
    ellL(i)= ell(i) - stell(i)/sqrt(N)*1.95;
    ellU(i)= ell(i) + stell(i)/sqrt(N)*1.95;
end
plot(theta,ell, theta, ellL, theta, ellU)
```

```
function out=H1(X)
Path_1=X(:,1)+X(:,4);
Path_2=X(:,1)+X(:,3)+X(:,5);
Path_3=X(:,2)+X(:,3)+X(:,4);
Path_4=X(:,2)+X(:,5);
out=min([Path_1,Path_2,Path_3,Path_4],[],2);
```

Next, suppose that X_1 is simulated under an $\mathsf{Exp}(1/\theta_0)$ distribution instead, and that we wish to estimate how $\ell(\theta)$ behaves under general $\theta > 0$. The estimator is now

$$\widehat{\ell}(\theta;\theta_0) = \frac{1}{N}\sum_{k=1}^{N} H(\mathbf{X}_k)\frac{\mathrm{e}^{-X_{k1}/\theta}/\theta}{\mathrm{e}^{-X_{k1}/\theta_0}/\theta_0} = \frac{\theta_0}{\theta\, N}\sum_{k=1}^{N} H(\mathbf{X}_k)\,\mathrm{e}^{X_{k1}(1/\theta_0-1/\theta)}\,. \qquad (9.50)$$

A typical estimate for the response curve for the case $N = 10^4$ and $\theta_0 = 3$ is depicted in Figure 9.7. The MATLAB code for this exponential case is available on the Handbook website.



Two significant differences with the uniform case are that (1) the estimated response curve is a smooth function of θ, and (2) it is possible to obtain estimates for all $\theta > 0$. However, as noted above, when θ is too large both the estimate and the confidence interval are unreliable. This is illustrated by the fact that the true response function must be monotone increasing in θ, whereas the estimate is decreasing from about $\theta > 7$ onward. It is not difficult to show, see [18, Page 210], that the variance of the estimator is infinite for $\theta > 2\theta_0 = 6$. Thus, it is not recommended to estimate $\ell(\theta)$ in this way for θ larger than 5, say. Note that for $\theta > \theta_0$ the importance sampling pdf $f(\mathbf{x};\theta_0)$ has thinner tails (decays quicker) in the x_1 variable than $f(\mathbf{x};\theta)$.

Finally, we mention that it is not difficult in this particular case to estimate the *entire* response function using only a single simulation run without importance sampling. The idea is to write

$$\ell(\theta) = \mathbb{E}h(\mathbf{U};\theta) ,$$

where, in the uniform case, $h(\mathbf{U};\theta) = H(\theta U_1, 2U_2, 3U_3, 2U_4, U_5)$ and $U_1,\ldots,U_5 \overset{\text{iid}}{\sim} \mathsf{U}(0,1)$, as in Example 9.1. Thus $\ell(\theta)$ is simply estimated as

$$\frac{1}{N}\sum_{k=1}^{N} h(\mathbf{U}_k;\theta) , \quad \theta > 0 ,$$

from a single iid sample $\mathbf{U}_1,\ldots,\mathbf{U}_N \overset{\text{iid}}{\sim} \mathsf{U}(0,1)^5$.

9.8 QUASI MONTE CARLO

Quasi Monte Carlo provides a powerful way to estimate d-dimensional integrals of the form

$$\ell = \int h(\mathbf{u})\,\mathrm{d}\mathbf{u}$$

by means of the sample average

$$\frac{1}{N}\sum_{\mathbf{u}\in\mathcal{P}_N} h(\mathbf{u}) ,$$

☞ 25 where $\mathcal{P}_N$ is a set of N quasirandom points, as is explained in Chapter 2. Error estimates can be obtained by randomizing the point set via a *random shift* (Section 2.7) and producing K independent copies of the estimator (2.10). Significant variance reduction can be obtained in this way; see, for example, [12]. The general procedure is summarized as follows.

Algorithm 9.11 (Quasi Monte Carlo Estimation)

1. *Generate a quasi Monte Carlo point set* $\mathcal{P}_N = \{\mathbf{u}_j,\ j = 1,\ldots,N\}$.
2. *Generate independent random vectors* $\mathbf{Z}_1,\ldots,\mathbf{Z}_K \overset{\text{iid}}{\sim} \mathsf{U}(0,1)^d$.
3. *Form the shifted point sets* $\mathcal{P}_N^{(i)} = (\mathcal{P}_N + \mathbf{Z}_i) \mod 1,\ i = 1,\ldots,K$.

4. *Calculate*

$$\widehat{\ell}_i = \frac{1}{N} \sum_{\mathbf{u} \in \mathcal{P}_N^{(i)}} h(\mathbf{u}), \quad i = 1, \ldots, K\,.$$

5. *Estimate* ℓ *as* $\widehat{\ell} = \frac{1}{K}\sum_{k=1}^{K} \widehat{\ell}_k$ *and determine an approximate* $1-\alpha$ *confidence interval as*

$$\left(\widehat{\ell} - z_{1-\alpha/2}\frac{S}{\sqrt{K}},\ \widehat{\ell} + z_{1-\alpha/2}\frac{S}{\sqrt{K}}\right), \tag{9.51}$$

where z_γ *denotes the* γ*-quantile of the* $\mathsf{N}(0,1)$ *distribution and* S *is the sample standard deviation of* $\widehat{\ell}_1, \ldots, \widehat{\ell}_K$.

■ EXAMPLE 9.13 (Quasi Monte Carlo for the Bridge Network)

As the expectation ℓ in (9.2) involves a relatively low-dimensional problem ($d = 5$), quasi Monte Carlo integration is expected to work well here. The MATLAB program below implements Algorithm 9.11 using a Faure point set, constructed via the MATLAB function `faure.m` defined on Page 33. The number of replications K is chosen to be 20, in order to give reasonable estimates for the relative error. The size of each of the 20 shifted point sets is $N = 500$, so that the total number of function evaluations of $h(\mathbf{u})$ is 10^4. A typical estimate is $\widehat{\ell} = 0.9308$ with an estimated relative error (that is, $S/(\sqrt{K}\widehat{\ell})$ with S as in (9.51)) of 0.072% which, for this particular problem, is better than those that were obtained by the other variance reduction methods in this chapter.

```
%brigeQMC_faure.m
K = 20;
N = 10^4/K;
F = faure(5,5,N-1);
for i=1:K
    U(:,:,i) = mod(F + repmat(rand(1,5),N,1), 1);
end
for i=1:K
    y(i) = mean(h(U(1:N,:,i)));
end
ell = mean(y);            %estimate
se = std(y)/sqrt(K); %standard error
fprintf('ell=%g, percRE = %g  \n',ell, 100*se/ell);
```

To further demonstrate the accuracy of the quasi Monte Carlo program, a typical outcome for $N = 50000$ and $K = 20$ is $\widehat{\ell} = 0.929862$, with an estimated relative error of 0.0027%. Recall that the true value is 0.929861111....

Finally, Figure 9.8 depicts, for different (quasi)random point sets, the convergence behavior of the estimator $\widehat{\ell}$ in Algorithm 9.11 using $K = 40$ repetitions and a point set of size N ranging from 8 to 10^5. All Monte Carlo methods are repeated $K = 40$ times. It is clearly seen that CMC has a significantly larger standard error (that is, $S/\sqrt{K}$ with S as in (9.51)) over the whole range of N. Moreover, CMC decreases at a slower rate than the Sobol', Faure, and Korobov point sets. The latter three are comparable in performance for this example. We used an extensible Korobov point set with parameter $a = 14471$.

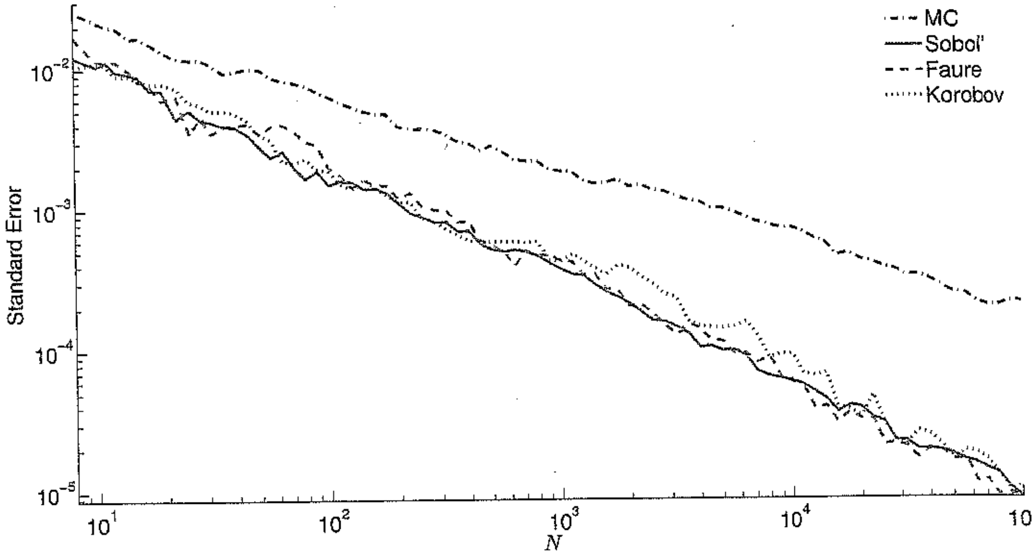


Figure 9.8 Standard error of the estimator $\widehat{\ell}$ with $K = 40$ versus the number of points N for different (quasi)random point sets.

In conclusion, in Figure 9.9 we summarize our *subjective* simulation experience of the different variance reduction techniques, as a guide for the practitioner. The figure indicates both the difficulty of implementation and the potential for improvement over CMC for each of the techniques. For completeness we include the ☞ 481 splitting method from Chapter 14.

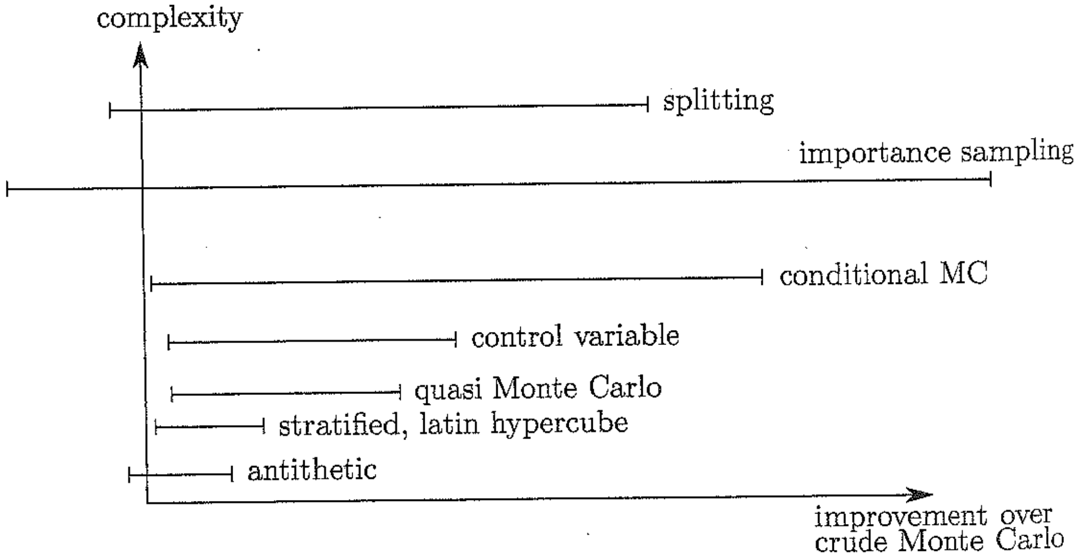


Figure 9.9 A guide for selecting a variance reduction technique.

Further Reading

The fundamental paper on variance reduction techniques is Kahn and Marshal [7]. There are many good Monte Carlo textbooks with chapters on variance reduction techniques. Among them are [1, 5, 6, 8, 9, 11, 13, 14, 15, 16, 20].

REFERENCES

1. S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis.* Springer-Verlag, New York, 2007.
2. S. Asmussen and R. Y. Rubinstein. Response surface estimation and sensitivity analysis via efficient change of measure. *Stochastic Models*, 9(3):313–339, 1993.
3. W. G. Cochran. *Sampling Techniques.* John Wiley & Sons, New York, third edition, 1977.
4. P. T. de Boer, D. P. Kroese, and R. Y. Rubinstein. A fast cross-entropy method for estimating buffer overflows in queueing networks. *Management Science*, 50(7):883–895, 2004.
5. G. S. Fishman. *Monte Carlo: Concepts, Algorithms and Applications.* Springer-Verlag, New York, 1996.
6. P. Glasserman. *Monte Carlo Methods in Financial Engineering.* Springer-Verlag, New York, 2004.
7. M. Kahn and A. W. Marshall. Methods of reducing sample size in Monte Carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.
8. J. P. C. Kleijnen. *Statistical Techniques in Simulation, Part 1.* Marcel Dekker, New York, 1974.
9. J. P. C. Kleijnen. Analysis of simulation with common random numbers: A note on Heikes et al. (1976). *Simuletter*, 11(2):7–13, 1979.
10. D. P. Kroese, T. Taimre, Z. I. Botev, and R. Y. Rubinstein. *Solutions Manual to Accompany: Simulation and the Monte Carlo Method, Second Edition.* John Wiley & Sons, New York, 2007.
11. A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis.* McGraw-Hill, New York, third edition, 2000.
12. P. L'Ecuyer and C. Lemieux. Variance reduction via lattice rules. *Management Science*, 46(9):1214–1235, 2000.
13. J. S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer-Verlag, New York, 2001.
14. D. L. McLeish. *Monte Carlo Simulation and Finance.* John Wiley & Sons, New York, 2005.
15. C. P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer-Verlag, New York, second edition, 2004.
16. S. M. Ross. *Simulation.* Academic Press, New York, third edition, 2002.
17. R. Y. Rubinstein and D. P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning.* Springer-Verlag, New York, 2004.

18. R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method.* John Wiley & Sons, New York, second edition, 2007.

19. R. Y. Rubinstein and A. Shapiro. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization via the Score Function Method.* John Wiley & Sons, New York, 1993.

20. I. M. Sobol'. *A Primer for the Monte Carlo Method.* CRC Press, Boca Raton, FL, 1994.

21. F. Stern. Conditional expectation of the duration in the classical ruin problem. *Mathematics Magazine*, 48(4):200–203, 1975.