



Monte Carlo Simulation in Engineering

Mikael Amelin

Draft version

KTH Royal Institute of Technology
Electric Power Systems
Stockholm 2013

PREFACE

This compendium describes how Monte Carlo methods can be applied to simulate technical systems. The description covers background on probability theory and random number generation as well as the theory and practice of efficient Monte Carlo simulations. The core of the compendium is based on lectures that have been given at KTH for several years; however, the presentation here also includes more explanatory texts and exercises with solutions.

I would like to give a warm thank you to colleagues and students who have helped improve the contents of this compendium by asking questions, pointing out errors and suggesting additional topics.

Stockholm
August 2015

Mikael Amelin

CONTENTS

| | |
|---|------------|
| Preface | iii |
| Contents | v |
| 1 Introduction | 1 |
| 1.1 Brief History | 1 |
| 1.2 Problem Definition | 1 |
| 1.3 Notation | 5 |
| 2 Random Variables | 7 |
| 2.1 Probability Distributions | 7 |
| 2.1.1 Populations | 7 |
| 2.1.2 Other Common Definitions of Probability Distributions | 8 |
| 2.2 Statistical Properties | 10 |
| 2.3 Arithmetics of Random Variables | 14 |
| Exercises | 15 |
| 3 Random Numbers | 17 |
| 3.1 Pseudo-random Numbers | 17 |
| 3.2 Transformation of Random Numbers | 19 |
| 3.2.1 Independent Random Numbers | 19 |
| 3.2.2 Correlated Random Numbers | 20 |
| Exercises | 21 |
| 4 Simple Sampling | 25 |
| 4.1 Estimating Expectation Values | 25 |
| 4.2 Other Estimates | 29 |
| 4.3 Application | 30 |
| Exercises | 31 |
| 5 Variance Reduction Techniques | 33 |
| 5.1 Complementary Random Numbers | 33 |
| 5.1.1 Principle | 33 |
| 5.1.2 Application | 34 |
| 5.2 Dagger Sampling | 36 |
| 5.2.1 Principle | 36 |
| 5.2.2 Application | 37 |
| 5.3 Control Variates | 37 |
| 5.3.1 Principle | 37 |
| 5.3.2 Application | 37 |
| 5.4 Correlated Sampling | 38 |
| 5.4.1 Principle | 38 |
| 5.4.2 Application | 38 |

| | | |
|----------|---|-----------|
| 5.5 | Importance Sampling | 38 |
| 5.5.1 | Principle | 38 |
| 5.5.2 | Application | 39 |
| 5.6 | Stratified Sampling | 40 |
| 5.6.1 | Principle | 40 |
| 5.6.2 | Application | 42 |
| | Exercises | 44 |
| 6 | Efficient Monte Carlo Simulations | 55 |
| 6.1 | Mathematical Model | 55 |
| 6.2 | Choice of Simulation Method | 55 |
| 6.3 | Testing | 55 |
| A | Probability Distributions | 57 |
| B | Akabuga District Simulation | 59 |
| B.1 | Analysis of the System | 59 |
| B.2 | Simulation Code | 62 |
| C | Probabilistic Production Cost Simulation | 77 |
| | Further reading | 77 |
| | Solutions to Exercises | 79 |

INTRODUCTION

Monte Carlo methods refers to a class of methods to solve mathematical problems using random samples. A straightforward example is the computation of the expectation value of a random variable; instead of computing the expectation value according to the definition (which may involve solving complex integrals) we observe the behaviour of the random variable, i.e., we collect samples, and estimate its expectation value based on these samples. However, Monte Carlo methods may also be used for solving deterministic problems. This might seem odd at a first glance, but the idea is simply to find a random variable, the statistic properties of which is depending on the solution to the deterministic problem. An example of this would be an opinion poll. Assume that there is going to be an election. If we ignore that people may change their preferences over time and consider just one specific point of time then the share of voters who are planning to vote for a certain candidate is deterministic. However, in order to compute the true value of this share, we would have to ask all voters which candidate they favour. An alternative would be to ask a limited number of randomly chosen voters and use these samples to estimate the share of votes the candidate will obtain.

This compendium will describe how Monte Carlo methods can be used for simulation of various technical systems. The compendium includes many mathematical definitions and formulae, but it should be emphasised that this is not a mathematical textbook. The focus of the presentation will be how Monte Carlo methods can be applied to solve engineering problems; hence, the mathematics should be seen as a tool and not a topic in itself. This means that for example mathematical proofs will only be provided in order to improve the understanding of the described methods, and some mathematical details might be ignored.

1.1 Brief History

The phrase “Monte Carlo methods” was coined in the beginning of the 20th century, and refers to the famous casino in Monaco¹—a place where random samples indeed play an important role. However, the origin of Monte Carlo methods is older than the casino.

To be added: History of probability theory...

To be added: Bernouille, Poisson and the law of large numbers...

To be added: Buffon’s needle

To be added: Modern development...

1.2 Problem Definition

This entire compendium is focusing on methods for simulation of systems on one specific format. (This might seem as a large limitation, but the reader will soon see that a wide range of systems fit into this format.) An overview of this format is given in figure 1.1. The studied systems are modelled by a set of

1. It has been said that if Monte Carlo methods had been first explored today, they would have been referred to as “Las Vegas methods”.

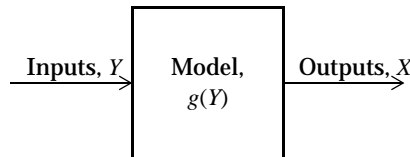


Figure 1.1 The simulation problem considered in this compendium.

random input variables, which we at this point simply collect into a vector, Y . The probability distribution of these inputs must be known. We also have and a set of output variables, which we also collect into a vector, X . As these outputs are depending on the random inputs, they must be random variables as well; however, the probability distribution of the outputs is not known—in fact, the objective of simulating the system is to determine the behaviour of the outputs. Finally, we have a mathematical model of the system, which determines how the values of the outputs are calculated given the values of the input variables. We denote the mathematical model as a function g , such that $X = g(Y)$. This function defines how the values of the outputs are computed given the values of the input values. In this compendium, a set of random values for each input will be referred to as a *scenario*.

Example 1.1. To be added: Example of a model expressed explicitly...

To be added: Discussion of model constants...

In example 1.1 it was possible to write the output as an explicit function of the input. This is however a special case, and most models $g(Y)$ will be defined implicitly and may require several steps before the output values are computed. A typical example would be there the output values depend on the solution to an optimisation problem, and also require a few other calculations to obtain the final results. for example from the solution to an optimisation problem, as illustrated in the next example:

Example 1.2 (simple two-area power system). Consider a power system divided in two areas: a generation centre and a load centre. The generation centre is dominated by large renewable power plants, but there is also a small local load, whereas the load centre has most of the load in the system, but there is also some thermal power plants. The renewable power plants in the generation centre are assumed to have negligible variable operation costs and the risk of outages is also negligible. Moreover, the capacity of the renewable power plants is larger than the maximal local load; hence, the generation centre always has excess power to export to the load centre. However, the interconnection between the areas has a limited capacity and there is a risk of outages. There are also electrical losses on the interconnection; these losses are proportional to the square of the power injected into the interconnection. The variable costs of the thermal power plants in the load centre is assumed to be directly proportional to the power output (i.e., start-up costs and ramp rates etc. may be neglected) and there is a risk of outages in these units. Finally, the load in the two areas is varying randomly. It can be assumed that the load is described by one probability distribution for the total load of the system, and another probability distribution for the share of the load that is located in the main load centre.

The system is operated in such a manner that the first priority is to avoid load shedding (i.e., when load has to be disconnected because there is not sufficient generation capacity available in the system) and the second priority is to minimise the generation costs. Voltage and frequency control may be neglected. The objective of simulating this power system is to determine the total operation cost and the risk of load shedding. Formulate a model of this kind of power system and show how the values of the outputs are calculated given the values of the inputs.

Solution: Start by introducing symbols for the inputs and outputs of this simulation problem:

Inputs

The inputs are the random variables of the model. In this case we have random outages in thermal power plants and the interconnection between the areas, as well as random loads in the two areas:

- c = share of the total load that is located in the main load centre,
- D_{tot} = total load in the system,
- \bar{G}_g = available generation capacity in thermal plants g ,
- \bar{P} = available transmission capacity on the interconnection between the two areas.

Outputs

We are actually only interested in two outputs (the total operation cost and whether load shedding occurs or not), but in order to compute these two values, we will need some partial results. Since the partial results also depend on the values of the random inputs, they will in practice also be outputs (i.e., random variables with unknown probability distributions depending on the probability distributions of the inputs). In this case, the model will be generating the following outputs:

- D_1 = load in the main load centre,
- D_2 = load in the generation centre,
- G_g = generation in thermal power plant g ,
- H = generation in the renewable power plants,
- $LOLO$ = loss of load occasion (binary variable equal to 1 if load shedding is necessary and 0 otherwise),
- P = transmission from the generation centre to the load centre,
- TOC = total operation cost,
- U = unserved load,

where $LOLO$ and TOC are the outputs that we want to study.

Model constants

In order to formulate the mathematical model, we are going to need some additional values:

- β_{Gg} = variable operation cost of thermal power plant g ,
- β_U = penalty cost for unserved load,
- γ = loss coefficient for the interconnection between the areas,
- \bar{H} = available generation capacity in the renewable power plants.

To be added: Comment the penalty cost for unserved load...

Mathematical model

Now we can formulate the calculations necessary to compute the values of the outputs for a scenario. First, we need to compute the local load in each area:

$$D_1 = c \cdot D_{tot} \tag{1.1a}$$

$$D_2 = (1 - c) \cdot D_{tot} \tag{1.1b}$$

The next step is to determine how the system will be operated. This can be formulated as an optimisation problem, where the objective function (1.2a) states that the costs of the system should be minimised, the constraints (1.2b), (1.2c) state that there should be balance between generation, load and import in the load centre and between generation, load and export in the generation centre and the limits (1.2d)–(1.2g) state that generation and transmission may not exceed the available capacity.

$$\text{minimise } \sum_g \beta_{Gg} G_g + \beta_U U, \tag{1.2a}$$

$$\text{subject to } \sum_g G_g + P - \gamma P^2 = D_1 - U, \tag{1.2b}$$

$$H = D_2 + P, \tag{1.2c}$$

$$0 \leq G_g \leq \bar{G}_g, \forall g, \tag{1.2d}$$

$$0 \leq H \leq \bar{H}, \tag{1.2e}$$

$$0 \leq P \leq \bar{P}, \tag{1.2f}$$

$$0 \leq U. \tag{1.2g}$$

To be added: Explanation of the model and description of how the problem can be solved using a simple algorithm...

Finally, once it has been determined how the power system is operated, the two main outputs can be computed from the solution to the optimisation problem:

$$LOLO = \begin{cases} 0 & \text{if } U = 0, \\ 1 & \text{if } U > 0, \end{cases} \tag{1.3a}$$

$$TOC = \sum_g \beta_{Gg} G_g. \tag{1.3b}$$

It is important to notice that the model g is deterministic! Hence, if we have two scenarios, y_1 and y_2 , producing two sets of output values, $x_1 = g(y_1)$ and $x_2 = g(y_2)$ then if $y_1 = y_2$ we will get that $x_1 = x_2$. If this property is not fulfilled, the model is missing inputs and should be reformulated, as in the following example:

Example 1.3. To be added: Example of a model which is missing an input value (power grid, where the reliability is depending on the success of reclosing breakers after a failure)

...

Many of the examples in the following chapters of this compendium will be based on a system based on the model from example 1.1. Additional details on this example system are collected in appendix B. Here, we will just introduce the system and identify the model constants:

Example 1.4 (Akabuga District). Akabuga is a small town in East Africa. The town is not connected to the national grid, but has a local system of its own. The local grid is supplied by a 350 kW hydro power plant in Ekyaalo. The variable costs of the hydro power plant are negligible and it can be considered to be 100% reliable. There is a 11 kV transmission line between Ekyaalo and Akabuga. This line has a maximal capacity of 300 kW, the reliability is 99% and the losses of the line are equal to $5 \cdot 10^{-5} P^2$, where P is the power injected on the Ekyaalo side. In Akabuga proper, there are two diesel generator sets. The first unit has 200 kW capacity, the variable costs are 10 ¢/kWh and the availability is 90%. The second unit has 150 kW capacity, the variable costs are 12 ¢/kWh and the availability is 80%.

The total load of the system is varying between 200 and 600 kW according to the probabilities listed in table 1.1. It is 50% probability that 85% of the total load is in Akabuga and 50% probability that 90% of the total load is in Akabuga.

State the values of the model constants for a simulation of Akabuga District.

Table 1.1 Probabilities of different load levels in Akabuga District.

| Total load [kW] | Probability [%] |
|-----------------|-----------------|
| 200 | 20 |
| 300 | 40 |
| 400 | 25 |
| 500 | 10 |
| 600 | 5 |

Solution: The model constants are the inputs that do not change value between scenarios. From the description above, we can identify the following model constants:

$$\begin{aligned} \beta_{G1} &= \text{variable operation cost of the large diesel generator set} = 10, \\ \beta_{G2} &= \text{variable operation cost of the small diesel generator set} = 12, \\ \gamma_L &= \text{loss coefficient for the 11 kV transmission line} = 5 \cdot 10^{-5}, \end{aligned}$$

\bar{H} = available generation capacity in the hydro power plant.

The capacity of diesel generator sets and the transmission line which are mentioned above are also not changing between scenarios. However, due to the risk of outages, the *available capacity* (which is what needs to be considered in the simulation) is a random variable; hence, the maximal capacity will appear in the probability distribution of the inputs (cf example 2.2).

1.3 Notation

Before we start investigating the application of Monte Carlo methods to solve the simulation problem described above, it might be useful to introduce a general notation, which will be used throughout this compendium. Once the reader is familiar with this notation, it will be more straightforward to interpret the mathematical expression appearing in the following chapters.

- **Random variables.** All random variables are denoted by upper-case Latin letters; usually just one single letter, for example Y or X , but sometimes several letters, such as $LOLO$ and TOC in example 1.2.
- **Samples.** An observation of a random variable (i.e., a sample) is denoted by the lower-case of the symbol used for the random variable itself, for example y or x . In most cases, we also use an index in order to separate different samples from each other, i.e., y_i or x_i .
- **Populations.** A population is denoted by the same upper-case Latin letter as the corresponding random variable, but using a script font, for example \mathcal{Y} or \mathcal{X} . The value of the i :th unit in a population is denoted by the lower-case symbol used for population itself, indexed by i , for example y_i or x_i .
- **Probability distributions.** Probability distributions are denoted by Latin f in upper or lower case (depending on interpretation²) and an index showing to which random variable the distribution is associated, for example f_Y or F_X . The idea of the index is to tell different probability distributions apart from each other.
- **Statistical properties.** Key statistical properties of a probability distribution are denoted by lower-case Greek letters and an index showing to which random variable the statistical property is associated, for example μ_Y or σ_X . The idea of the index is to tell different probability distributions apart from each other.
- **Estimates.** Estimates of key statistical properties for a probability distribution are denoted by upper or lower case (depending on interpretation³) Latin counterpart of the symbol used for the statistical property itself and an index showing to which random variable the statistical property is associated, for example M_X or s_X . The idea of the index is to tell different probability distributions apart from each other.

2. Cf. section 2.1.

3. Cf. section 4.1.

RANDOM VARIABLES

As the idea of Monte Carlo simulation is to estimate values using random observations, it is natural that a basic understanding of probability theory is necessary. This chapter summarises the probability theory that will be used in the remainder of the compendium. The focus of the presentation will be on random variables.

2.1 Probability Distributions

Intuitively, we may understand a random variable exactly as the name suggests, i.e., a variable, the value of which is varying according to some random pattern. This pattern, which characterises the behaviour of the random variable is referred to as its probability distribution. There is an infinite number of possible probability distributions; however, some common classes of distributions have been identified and named. A brief overview can be found in appendix A.

2.1.1 Populations

The formal mathematical definition of a random variable is however slightly more complex, and will therefore not be discussed here.

For a discussion of sampling and Monte Carlo simulation, a useful interpretation of random variables is to consider a random variable to be associated with a certain *population*, which we define as follows:

Definition 2.1. The random variable X corresponds to a population, \mathcal{X} , which is a set with N members (which are referred to as “units”). Each unit has a value, x_i , which may be multi-dimensional. The values of the units in \mathcal{X} do not have to be unique, but they should include all possible outcomes of the random variable X and the relative occurrence of a certain value should be proportional to the probability of the corresponding outcome.

Example 2.1. State the population corresponding to the following random variables:

a) D , which represents the result of throwing a normal six-sided dice.

b) To be added...

Solution:

a) $\mathcal{D} = \{1, 2, 3, 4, 5, 6\}$

b) To be added...

Based on this definition we can distinguish between some main categories of populations (and consequently between different categories of random variables). First we can differentiate between variables where the outcome can only belong to specific, discrete values or if the outcome can be found in continuous intervals:

Definition 2.2. If the population is finite or countable infinite, the random variable is *dis-*

crete; otherwise, it is *continuous*.

To be added: Examples...

As pointed out in definition 2.1, the units in a population may have more than one value. If each unit has one value, the population directly corresponds to one random variable. However, if the units have more than one value, we may consider each value to represent a separate random variable, which then have a joint probability distribution.

Definition 2.3. If each unit in the population is associated to a single value, the probability distribution is *univariate*; otherwise it is *multi-variate*.

To be added: Examples...

Finally, we can also study how the values of the units in a population is varying:

Definition 2.4. If all units in univariate population have the same or almost the same value, the population is said to be *homogeneous*.

Definition 2.5. If most units in univariate population have the different values, the population is said to be *heterogeneous*.

Definition 2.6. If the majority of the units in a univariate population have the same value (these units are referred to as the *conformist* units), the population is said to be *duogeneous*. The remainder of the population (which is referred to as the *diverging* units) may either be homogeneous (i.e., all diverging units have the same value) or heterogeneous (i.e., the diverging units have different values).

To be added: Examples...

It may be noted that the difference between a homogeneous and a heterogeneous population may depend on the situation.

To be added: Example where the population is homogeneous for a rough estimate, whereas it can be considered heterogeneous if a precise estimate is required.

2.1.2 Other Common Definitions of Probability Distributions

There are several ways to represent probability distributions mathematically. Populations are often useful for describing and understanding Monte Carlo simulation, but other definitions can also be useful in some cases, and are also frequently applied in other fields of statistical analysis and stochastic methods. Hence, it is important to be familiar with the following definitions:

Definition 2.7. The probability that an observation of a univariate, discrete random variable X is equal to the value x is given by the frequency function, $f_X(x)$, i.e.,

$$P(X = x) = f_X(x).$$

If we compare the definition of a frequency function to a population, we see that the value of the frequency function is equal to the number of units in the population having the value x compared to the total size of the population, i.e.,

$$f_X(x) = \frac{N_{X=x}}{N}. \tag{2.1}$$

A similar definition is used for continuous random variables. However, as the probability of getting exactly the value x is infinitesimal, the function $f_X(x)$ is referred to as a density function¹ and represents the probability that an observation is within a given range:

Definition 2.8. The probability that an observation of a univariate, continuous random variable X belongs to a set \mathcal{X} is given by the density function, $f_X(x)$, i.e.,

$$P(X \in \mathcal{X}) = \int_{\mathcal{X}} f_X(x) dx.$$

By definition, all units in a population has a value. This means that frequency and density functions

1. This denomination is sometimes also used for discrete random variables; a frequency function can be considered as a special type of density function.

must fulfil the following:

$$\sum_{x=-\infty}^{\infty} f_X(x) = 1 \text{ (discrete random variable),} \quad (2.2a)$$

$$\int_{-\infty}^{\infty} f_X(x)dx = 1 \text{ (continuous random variable).} \quad (2.2b)$$

An alternative to density functions is to study distribution functions:

Definition 2.9. The probability that an observation of a univariate, discrete random variable X is less than or equal to the value x is given by the distribution function, $F_X(x)$, i.e.,

$$P(X \leq x) = F_X(x).$$

From definition 2.9 it follows that $F_X(x)$ must be an increasing function,² that $F_X(x) \rightarrow 0$ if $x \rightarrow -\infty$ and that $F_X(x) \rightarrow 1$ if $x \rightarrow +\infty$. Moreover, comparing definitions 2.8 and 2.9, we see that

$$P(X \leq x) = F_X(x) = \int_{-\infty}^x f_X(x)dx, \quad (2.3)$$

i.e., the distribution function is the primitive function of the density function. (This is the reason for choosing the symbols F and f for distribution function and density function analogous to the notation used in calculus.)

Finally, in some applications it is preferable to consider a duration curve instead of the distribution function. (An example is simulation of electricity markets, cf. appendix C.) Duration curves can be designed in different manners, but the one most suitable for probability analysis is the normalised inverse duration curve, which we in this compendium will refer to simply as a duration curve.

Definition 2.10. The probability that an observation of a univariate, discrete random variable X is larger than the value x is given by the duration curve, $\tilde{F}_X(x)$, i.e.,

$$P(X > x) = \tilde{F}_X(x).$$

In a similar way as for distribution functions, we can notice that $\tilde{F}_X(x)$ must be an increasing function, that $\tilde{F}_X(x) \rightarrow 1$ if $x \rightarrow -\infty$, that $\tilde{F}_X(x) \rightarrow 0$ if $x \rightarrow +\infty$ and that

$$P(X > x) = \tilde{F}_X(x) = 1 - F_X(x) = \int_x^{\infty} f_X(x)dx. \quad (2.4)$$

To be added: Comments on multivariate distributions...

Let us now express the probability distributions of the inputs in the Akabuga District example using the definitions above:

Example 2.2 (Akabuga District). Akabuga is a small town in East Africa. The town is not connected to the national grid, but has a local system of its own. The local grid is supplied by a 350 kW hydro power plant in Ekyaalo. The variable costs of the hydro power plant are negligible and it can be considered to be 100% reliable. There is a 11 kV transmission line between Ekyaalo and Akabuga. This line has a maximal capacity of 300 kW, the reliability is 99% and the losses of the line are equal to $5 \cdot 10^{-5}P^2$, where P is the power injected on the Ekyaalo side. In Akabuga proper, there are two diesel generator sets. The first unit has 200 kW capacity, the variable costs are 10 ¢/kWh and the availability is 90%. The second unit has 150 kW capacity, the variable costs are 12 ¢/kWh and the availability is 80%.

The total load of the system is varying between 200 and 600 kW according to the probabilities listed in table 2.1. It is 50% probability that 85% of the total load is in Akabuga and 50% probability that 90% of the total load is in Akabuga.

2. The probability that a random variable is less than or equal to a cannot be smaller than the probability that a random variable is less than or equal to b if $b > a$.

State the frequency functions of the inputs in a simulation of Akabuga District.

Table 2.1 Total load in Akabuga District.

| | | | | | |
|-----------------|-----|-----|-----|-----|-----|
| Load [kW] | 200 | 300 | 400 | 500 | 600 |
| Probability [%] | 20 | 40 | 25 | 10 | 5 |

Solution: The frequency function states the probability of a certain outcome; thus, from the description above, we can identify the following frequency functions:

$f_{\bar{G}_1}$ = available generation capacity in the large diesel generator set =

$$= \begin{cases} 0.1 & x = 0, \\ 0.9 & x = 200, \\ 0 & \text{all other } x, \end{cases}$$

$f_{\bar{G}_2}$ = available generation capacity in the large diesel generator set =

$$= \begin{cases} 0.2 & x = 0, \\ 0.8 & x = 150, \\ 0 & \text{all other } x, \end{cases}$$

$f_{\bar{P}}$ = available generation capacity in the large diesel generator set =

$$= \begin{cases} 0.01 & x = 0, \\ 0.99 & x = 300, \\ 0 & \text{all other } x, \end{cases}$$

$$f_{D_{tot}} = \text{total demand} = \begin{cases} 0.2 & x = 200, \\ 0.4 & x = 300, \\ 0.25 & x = 400, \\ 0.1 & x = 500, \\ 0.05 & x = 600, \\ 0 & \text{all other } x, \end{cases}$$

$$f_C = \text{the share of the total demand located in Akabuga} = \begin{cases} 0.5 & x = 0.85, \\ 0.5 & x = 0.9, \\ 0 & \text{all other } x. \end{cases}$$

2.2 Statistical Properties

The probability distribution of a random variable is a complete description of its behaviour. However, in many cases we do not need such detailed information, but would prefer some key values that describe the main characteristics of the random variable. Therefore, different statistical measures have been introduced. The most important statistical measures are defined below.

Expectation Value

The expectation value of random variable is a the mean of all possible outcomes weighted by probability:

Definition 2.11. The expectation value of a random variable X is given by

$$E[X] = \frac{1}{N} \sum_{i=1}^N x_i \text{ (population),}$$

$$E[X] = \sum_{x \in \Omega_X} f_X(x)x \text{ (discrete random variable),}$$

$$E[X] = \int_{x \in \Omega_X} f_X(x)x dx \text{ (continuous random variable).}$$

As seen above, the definition varies slightly depending on whether the probability distribution is expressed as a population or using a density function. We may notice that it is the expression corresponding to the weighting according to probability that varies; hence, the following parts in the definitions above fulfil the same purpose:

$$\frac{1}{N} \sum_{i=1}^N \leftrightarrow \sum_{x \in \Omega_X} f_X(x) \leftrightarrow \int_{x \in \Omega_X} f_X(x) \dots dx.$$

As we will see below, this pattern will appear also in the other definitions of statistical properties.

The practical interpretation of the expectation value is that if we have a set of samples of a random variable, and this set is distributed exactly according to the probability distribution, then the expectation value is the mean of those samples. We can intuitively understand that if we have a large number of samples, it is quite likely that the samples will be distributed almost according to the probability distribution of the variable; hence, the mean of a large number of samples should be approximately equal to the expectation value. (In fact, this is the very foundation of simple sampling, as we will see in chapter 4).

Although most random variables have a well-defined expectation value, one should be aware that there is no guarantee that this is the case. This might seem strange, and is best understood by an example:

Example 2.3 (The St Petersburg paradox). Consider a game where a player pays a fixed fee to participate. The player then tosses a coin until a head appears. If a head appears in the j :th trial, the payout of the game is 2^j .

To be added: Expected payout of the game...

Variance and Standard Deviation

The variance of a random variable describes how much a random variable is varying around the expectation value.

To be added: Figure

Definition 2.12. The variance of a random variable X is given by

$$\text{Var}[X] = E[(X - E[X])^2] = E[X^2] - (E[X])^2 \text{ (general definition),}$$

$$\text{Var}[X] = \frac{1}{N} \sum_{i=1}^N (x_i - E[X])^2 \text{ (population),}$$

$$\text{Var}[X] = \sum_{x \in \Omega_X} f_X(x)(x - E[X])^2 \text{ (discrete random variable),}$$

$$\text{Var}[X] = \int_{x \in \Omega_X} f_X(x)(x - E[X])^2 dx \text{ (continuous random variable).}$$

A disadvantage with variance is that the unit of $\text{Var}[X]$ is the square of the unit of X . For example, if X is the unserved load in a power system and expressed in kWh/h then the variance of X is expressed in $(\text{kWh/h})^2$. In many cases, it is more convenient to have a measure of the variation that is directly comparable to the variable itself and the expectation value. Therefore, the notion of standard deviation has been introduced:

Definition 2.13. The standard deviation of a random variable X is given by

$$\sigma_X = \sqrt{\text{Var}[X]}.$$

Covariance and Correlation Coefficient

The covariance describes how different random variables in a multivariate distribution are interacting with each other:

Definition 2.14. The covariance of two random variables X and Y is given by

$$\text{Cov}[X, Y] = E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y].$$

Switching place between X and Y in definition 2.14 does not change the result, i.e.,

$$\text{Cov}[X, Y] = \text{Cov}[Y, X]. \tag{2.5}$$

We can also observe that the covariance between a variable and itself is equal to the variance of the variable, i.e.,

$$\text{Cov}[X, X] = \text{Var}[X]. \tag{2.6}$$

To be added: Discussion of covariance matrix...

Definition 2.15. A covariance matrix shows the covariances between all random variables in a multivariate probability distribution:

$$\Sigma_X = \begin{bmatrix} \text{Var}[X_1] & \text{Cov}[X_1, X_2] & \dots & \text{Cov}[X_1, X_k] \\ \text{Cov}[X_2, X_1] & \text{Var}[X_2] & & \\ \vdots & & \ddots & \\ \text{Cov}[X_k, X_1] & & & \text{Var}[X_k] \end{bmatrix}.$$

To be added: Discussion of correlation coefficient...

Definition 2.16. The correlation factor of two random variables X and Y is given by

$$\rho_{X, Y} = \frac{\text{Cov}[X, Y]}{\sqrt{\text{Var}[X]\text{Var}[Y]}}.$$

Example 2.4 (load in Akabuga and Ekyaalo). Consider Akabuga District (cf. example 1.4). What is the correlation coefficient between the load in Akabuga and Ekyaalo?

Solution: Let D_1 and D_2 denote the load in Akabuga and Ekyaalo respectively. These random variables are direct functions of the total load, D_{tot} and the share of the total load that is located in Akabuga, C , which have known probability distributions (see example 2.2). All possible results are listed in table 2.2. From the table, we can now compute

$$E[D_1 D_2] = 13\,811.25,$$

$$E[D_1] = 297.5,$$

$$E[D_2] = 42.5,$$

which results in

$$\text{Cov}[D_1, D_2] = E[D_1 D_2] - E[D_1]E[D_2] = 13\,811.25 - 297.5 \cdot 42.5 = 1\,167.5.$$

To compute the correlation coefficient we also need the variances of the area loads, which according to the definition and the probabilities listed in table 2.2 yields

$$\text{Var}[D_1] = 8\,807.5,$$

$$\text{Var}[D_2] = 257.5.$$

Now the correlation coefficient is given by

$$\rho_{D_1, D_2} = \frac{\text{Cov}[D_1, D_2]}{\sqrt{\text{Var}[D_1]\text{Var}[D_2]}} = \frac{1\,167.5}{\sqrt{8\,807.5 \cdot 257.5}} \approx 0.78,$$

i.e., a rather strong positive correlation.

Table 2.2 Sample space for the load in Akabuga and Ekyaaloo.

| D_{tot} | C | D_1 | D_2 | D_1D_2 | Probability, $f_{D_{tot}} \cdot f_C$ |
|-----------|------|-------|-------|----------|---|
| 200 | 0.85 | 170 | 30 | 5 100 | 0.1 |
| 200 | 0.9 | 180 | 20 | 3 600 | 0.1 |
| 300 | 0.85 | 255 | 45 | 11 475 | 0.2 |
| 300 | 0.9 | 270 | 30 | 8 100 | 0.2 |
| 400 | 0.85 | 340 | 60 | 20 400 | 0.125 |
| 400 | 0.9 | 360 | 40 | 14 400 | 0.125 |
| 500 | 0.85 | 425 | 75 | 31 875 | 0.05 |
| 500 | 0.9 | 450 | 50 | 22 500 | 0.05 |
| 600 | 0.85 | 510 | 90 | 45 900 | 0.025 |
| 600 | 0.9 | 540 | 60 | 32 400 | 0.025 |

To be added: Discussion of independent random variables...

Definition 2.17. X and Y are independent random variables if it holds for each x and y that

$$f_{X, Y}(x, y) = f_X(x)f_Y(y) \Leftrightarrow F_{X, Y}(x, y) = F_X(x)F_Y(y).$$

Theorem 2.18. If X and Y are independent then

$$E[XY] = E[X]E[Y].$$

Corollary 2.19. If X and Y are independent then they are also uncorrelated.

It should be noted that the reverse of corollary 2.19 does not hold, i.e., if two variables are uncorrelated we cannot conclude that they are independent! This is because the correlation is only a measure of the linear dependence of two random variables; hence, if there is a purely non-linear relation between two random variables, they will still be uncorrelated. Compare the following two examples:

Example 2.5. Assume that Y has the frequency function

$$f_X(x) = \begin{cases} 1/5 & \text{if } x = 0, 1, 2, 3, 4, \\ 0 & \text{otherwise.} \end{cases}$$

and assume that $X = Y^2$. Are X and Y uncorrelated?

Solution: We start by computing expectation values according to definition 2.11:

$$E[XY] = \frac{1}{5}(0 + 1^2 \cdot 1 + 2^2 \cdot 2 + 3^2 \cdot 3 + 4^2 \cdot 4) = 20,$$

$$E[X] = \frac{1}{5}(0 + 1^2 + 2^2 + 3^2 + 4^2) = 6,$$

$$E[Y] = \frac{1}{5}(0 + 1 + 2 + 3 + 4) = 2.$$

The covariance of X and Y can now be computed according to definition 2.14:

$$\text{Cov}[X, Y] = E[XY] - E[X]E[Y] = 20 + 6 \cdot 2 = 32.$$

As the covariance is larger than zero, we can conclude that X and Y are positively correlated.

Example 2.6. Assume that Y has the frequency function

$$f_X(x) = \begin{cases} 1/5 & \text{if } x = -2, -1, 0, 1, 2, \\ 0 & \text{otherwise.} \end{cases}$$

and assume that $X = Y^2$. Are X and Y uncorrelated?

Solution: We start by computing expectation values according to definition 2.11:

$$E[XY] = \frac{1}{5}((-2)^2 \cdot (-2) + (-1)^2 \cdot (-1) + 0 + 1^2 \cdot 1 + 2^2 \cdot 2) = 0,$$

$$E[X] = \frac{1}{5}((-2)^2 + (-1)^2 + 0 + 1^2 + 2^2) = 2,$$

$$E[Y] = \frac{1}{5}((-2) + (-1) + 0 + 1 + 2) = 0.$$

The covariance of X and Y can now be computed according to definition 2.14:

$$\text{Cov}[X, Y] = E[XY] - E[X]E[Y] = 0 + 2 \cdot 0 = 0.$$

As the covariance is zero, we can conclude that X and Y are uncorrelated.

2.3 Arithmetics of Random Variables

If a random variable X is a function of another random variable Y then it is generally difficult to directly compute the probability distribution of X based on the probability distribution of Y —this is one of the reasons why we would like to apply Monte Carlo simulation instead. However, there are some specific cases, where direct computations are straightforward. Some of these cases are described in the following theorems and will be used in the theoretical analysis of Monte Carlo simulation in this compendium.

Theorem 2.20. (Calculation of expectation value) Assume that a is a scalar, and Y, Y_1 and Y_2 are random variables. The following rules then apply to expectation values:

i) $E[aY] = aE[Y],$

ii) $E[Y_1 + Y_2] = E[Y_1] + E[Y_2],$

iii) $E[g(Y)] = \frac{1}{N} \sum_{i=1}^N g(y_i)$ (population),

$$E[g(Y)] = \sum_{y \in \Omega_Y} f_Y(y)g(y)$$
 (discrete random variable),

$$E[g(Y)] = \int_{(y \in \Omega_Y)} f_Y(y)g(y)dy$$
 (continuous random variable).

Theorem 2.21. (Calculation of variance) Assume that a is a scalar, and Y, Y_1, Y_2 etc. are random variables. The following rules then apply to variances:

i) $\text{Var}[aY] = a^2\text{Var}[Y],$

ii) $\text{Var}[Y_1 + Y_2] = \text{Var}[Y_1] + \text{Var}[Y_2] + 2\text{Cov}[Y_1, Y_2],$

iii) $\text{Var}[Y_1 - Y_2] = \text{Var}[Y_1] + \text{Var}[Y_2] - 2\text{Cov}[Y_1, Y_2],$

iv) $\text{Var}\left[\sum_{i=1}^k Y_i\right] = \sum_{i=1}^k \sum_{j=1}^k \text{Cov}[Y_i, Y_j].$

Theorem 2.22. (Convolution) If Y_1 and Y_2 are independent random variables then the probability distribution of $X = Y_1 + Y_2$ can be computed using convolution:

i) $f_X(x) = \sum_t f_{Y_1}(t)f_{Y_2}(x-t)$ (discrete random variables),

$$\text{ii) } f_X(x) = \int_{-\infty}^{\infty} f_{Y_1}(t)f_{Y_2}(x-t) \text{ (continuous random variables).}$$

Exercises

2.1 Are the following random variables heterogeneous or duogeneous?

$$\text{a) } f_X(x) = \begin{cases} 0.1 & \text{if } 10 \leq x \leq 20, \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{b) } f_X(x) = \begin{cases} 1/11 & \text{if } x = 10, 11, \dots, 20, \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{c) } f_X(x) = \begin{cases} 0.6 & \text{if } x = 10, \\ 0.4 & \text{if } x = 20, \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{d) } f_X(x) = \begin{cases} 0.6 & \text{if } x = 10, \\ 0.04 & \text{if } 10 < x \leq 20, \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{e) } f_X(x) = \begin{cases} 0.3 & \text{if } 10 \leq x \leq 12, \\ 0.05 & \text{if } 12 < x \leq 20, \\ 0 & \text{otherwise.} \end{cases}$$

More exercises to be added...

RANDOM NUMBERS

The idea of a Monte Carlo simulation is to estimate the behaviour of the simulated system using random samples. When sampling a physical system, the randomness of the samples will be generated by the system itself. However, when simulating a system using a mathematical model, it will be necessary to generate random input values. This chapter will present methods how to do this. The presentation will start with a description how to generate $U(0, 1)$ -distributed random numbers and then it is explained how random numbers of any other distribution can be obtained by transformation of random numbers from a $U(0, 1)$ -distribution.

3.1 Pseudo-random Numbers

One possibility to provide random numbers in a computer simulation would be to use some kind of hardware device. Such a device could be designed to generate truly random values, but there would also be an important disadvantage, namely that we would lack control of the produced random numbers. As a consequence, running the same simulation twice would generally not produce the same set of samples, which can be a problem especially when testing a simulation method or model. For example, assume that we run a simulation and we detect a few scenarios where the mathematical model produces erroneous output models. Once the errors have been corrected, it would be practical to be able to run the same scenarios again in order to verify that the problem has been solved.

Therefore, random number generation in computers are based on special mathematical functions or algorithms, which given one or more initial values (referred to as *seeds*) produces a sequence of numbers between 0 and 1. This sequence is in reality deterministic, which means that if the same seed is used, it will produce the same sequence (which makes simulations repeatable). Since these functions are not truly random, they are called *pseudo-random number generators*. However, a properly designed function will generate a sequence that has properties as close as possible to that of a true sequence of independent $U(0, 1)$ -distributed random numbers.

In practice, we do not need to worry about designing good pseudo-random number generators, as such functions are readily available in almost all high-level programming languages. In fact, it is preferable to use built-in pseudo-random number generators rather than programming one of your own, as the built-in functions should have been carefully designed to provide appropriate statistical properties.

Nevertheless, it can be of interest to get an idea of the principles for generation of pseudo-random numbers. As an example, we will study the widely used *linear congruential generator*. The sequence of numbers for this pseudo-random number generator is computed using the following formulae:

$$X_{i+1} = (aX_i + c) \bmod m, \quad (3.1a)$$

$$U_i = \frac{X_i}{m}, \quad (3.1b)$$

where

$$X_1 = \text{the seed},$$

- X_i = internal state i ,
- a = an integer multiplier ($0 < a < m$),
- c = an integer increment ($0 < c < m$),
- m = an integer divisor ($0 < m$),
- U_i = pseudo-random number i .

The modulo operator in (3.1a) returns the remainder when dividing $(aX_i + c)$ by m . The result of this operation is an integer in the interval $0, \dots, m - 1$. This means that the linear congruential generator can at most produce m possible values before the sequence starts repeating itself. In order to make the sequence as similar to a uniform distribution as possible, we would like the sequence to be as long as possible. This can be achieved if the following rules are considered when choosing the constants a , c and m :

- The greatest common divisor of c and m should be 1.
- $a - 1$ should be divisible by all prime factors of m .
- $a - 1$ should be a multiple of 4 if m is a multiple of 4.

The application of linear congruential generators is demonstrated in the following two examples:

Example 3.1. What sequence of numbers is generated by a linear congruential generator where $a = 5$, $c = 3$ and $m = 8$ for the seed $X_1 = 1$?

Solution: We can notice that the constants are fulfil the requirements above. The computations when starting with $X_1 = 1$ and applying (3.1a) are shown in table 3.1 below. We can see that the sequence will repeat itself after eight values, which is the maximal length of the sequence when $m = 8$.

Table 3.1 Computation of the random number sequence in example 3.1.

| | | | | | | | | |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|
| i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| X_i | 1 | 0 | 3 | 2 | 5 | 4 | 7 | 6 |
| U_i | 0.125 | 0.000 | 0.375 | 0.250 | 0.625 | 0.500 | 0.875 | 0.750 |
| $5X_i + 3$ | 8 | 3 | 18 | 13 | 28 | 15 | 38 | 33 |
| X_{i+1} | 0 | 3 | 2 | 5 | 4 | 7 | 6 | 1 |

Example 3.2. What sequence of numbers is generated by a linear congruential generator where $a = 5$, $c = 6$ and $m = 8$ for the seed $X_1 = 1$?

Solution: This time we do not fulfil the first requirement, as both c and m can be divided by 2. The computations when starting with $X_1 = 1$ and applying (3.1a) are shown in table 3.2 below. This time we only get four values before the sequence starts repeating itself, and for the remainder of the sequence we will only see the internal states 3, 5 and 7.

Table 3.2 Computation of the random number sequence in example 3.2.

| | | | | |
|------------|-------|-------|-------|-------|
| i | 1 | 2 | 3 | 4 |
| X_i | 1 | 3 | 5 | 7 |
| U_i | 0.125 | 0.375 | 0.625 | 0.750 |
| $5X_i + 6$ | 11 | 21 | 31 | 41 |
| X_{i+1} | 3 | 5 | 7 | 3 |

In an actual Monte Carlo simulation, it is always recommended to use the built-in pseudorandom number generator of the programming language used. However, in the examples in this compendium, it can be interesting to see how the random numbers were computed. Therefore, let us consider one last example of the linear congruential generator:

Example 3.3. Generate 20 numbers using a linear congruential generator where $a = 21$, $c = 3$ and $m = 1\,000$ for the seed $X_1 = 770$?

Solution: The first 20 values in the sequence when applying (3.1a) and (3.1b) are shown in the table below.

Table 3.3 Computation of the random number sequence in example 3.3.

| | | | | | | | | | | |
|-------------|--------|--------|--------|--------|--------|-------|-------|--------|--------|--------|
| i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| X_i | 770 | 173 | 636 | 359 | 542 | 385 | 88 | 851 | 874 | 357 |
| U_i | 0.770 | 0.173 | 0.636 | 0.359 | 0.542 | 0.385 | 0.088 | 0.851 | 0.874 | 0.357 |
| $21X_i + 3$ | 16 173 | 3 636 | 13 359 | 7 542 | 11 385 | 8 088 | 1 851 | 17 874 | 18 357 | 7 500 |
| X_{i+1} | 173 | 636 | 359 | 542 | 385 | 88 | 851 | 874 | 357 | 500 |
| i | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| X_i | 500 | 503 | 566 | 889 | 672 | 115 | 418 | 781 | 404 | 487 |
| U_i | 0.500 | 0.503 | 0.566 | 0.889 | 0.672 | 0.115 | 0.418 | 0.781 | 0.404 | 0.487 |
| $21X_i + 3$ | 10 503 | 10 566 | 11 889 | 18 672 | 14 115 | 2 418 | 8 781 | 16 404 | 8 487 | 10 230 |
| X_{i+1} | 503 | 566 | 889 | 672 | 115 | 418 | 781 | 404 | 487 | 230 |

3.2 Transformation of Random Numbers

As described in the previous section, random inputs for Monte Carlo simulations can be created by standard pseudorandom number generators available in almost all programming languages. However, these generators produce $U(0, 1)$ -distributed random numbers and unfortunately it is rarely so that all inputs of the system to be simulated have that probability distribution. Hence, we need to be able to convert $U(0, 1)$ -distributed random numbers to the actual probability distributions of the inputs. There are several methods that can be applied to perform this transformation. In this section, some methods that are well suited for Monte Carlo simulations are described.

The inputs to a Monte Carlo simulation do not need to be independent—the possibility to simulate systems with correlated inputs is one of the advantages of Monte Carlo simulation. However, generating random numbers for correlated random variables can be tricky, and therefore it is convenient if all inputs can be considered as independent. It might therefore be preferable to neglect weak correlations or to rearrange the inputs to avoid correlations.¹

3.2.1 Independent Random Numbers

Most probability distributions can be generated using the inverse transform method:

Theorem 3.1. (Inverse Transform Method) If U is a $U(0, 1)$ -distributed random number then Y is a distributed according to the distribution function $F_Y(x)$ if Y is calculated according to $Y = F_Y^{-1}(U)$.

To be added: Examples...

Random Numbers from Finite Populations

To be added: Introduction...

To be added: Search algorithm...

To be added: Comments... (Notice that the states must be sorted if complementary random numbers will be applied to an input!)

To be added: Examples...

Normally Distributed Random Numbers

To be added: Introduction...

1. The latter can be done by dividing the population in separate subpopulations, which are simulated separately (cf. section 5.6).

Theorem 3.2. (Approximate Inverse Transform Method) If U is a $U(0, 1)$ -distributed random number then Y is a $N(0, 1)$ -distributed random number, if Y is calculated according to

$$Q = \begin{cases} U & \text{if } 0 \leq U \leq 0.5, \\ 1 - U & \text{if } 0.5 < U \leq 1, \end{cases}$$

$$t = \sqrt{-2 \ln Q},$$

$$c_0 = 2.515517, \quad c_1 = 0.802853, \quad c_2 = 0.010328,$$

$$d_1 = 1.432788, \quad d_2 = 0.189269, \quad d_3 = 0.001308,$$

$$z = t - \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3}$$

and finally

$$Y = \begin{cases} -z & \text{if } 0 \leq U < 0.5, \\ 0 & \text{if } U = 0.5, \\ z & \text{if } 0.5 < U \leq 1. \end{cases}$$

To be added: Comments and examples...

3.2.2 Correlated Random Numbers

A Monte Carlo simulation can have inputs that are correlated. In some cases, generation of correlated random numbers is quite straightforward, but there is also a possibility that the transformation procedure becomes quite time-consuming. In the latter case, one should consider if very weak correlations should be ignored in the simulation or if the variables can be redefined using independent random variables instead.²

Below follows descriptions of methods to generate correlated random numbers.

Normally Distributed Random Numbers

It is convenient if any correlated random variables in a Monte Carlo simulation are normally distributed, because there is then a straightforward transformation from independent normally distributed random numbers (which can be generated using the approximative inverse transform method) to normally distributed random numbers with any correlation:

Theorem 3.3. Let $X = [X_1, \dots, X_K]^T$ be a vector of independent $N(0, 1)$ -distributed components. Let $B = \Sigma^{1/2}$, i.e., let λ_i and g_i be the i :th eigenvalue and the i :th eigenvector of Σ and define the following matrices:

$$P = [g_1, \dots, g_K],$$

$$A = \text{diag}(\lambda_1, \dots, \lambda_K),$$

$$B = P A^{1/2} P^T.$$

Then $Y = \mu + BX$ is a random vector where the elements are normally distributed with the mean μ and the covariance matrix Σ .

Random Numbers from Finite Populations

As for normally distributed random numbers, finite populations are a special case where generation of

2. Consider for example how the loads in Akabuga and Ekyaalo, which are strongly positively correlated (cf. example 2.4) can be replicated by two independent probability distributions for the total load and the share of the load that is located in Akabuga (cf. example 2.2).

correlated random is straightforward. The idea is to set up a univariate probability distribution of states. The inverse transform method is then applied to randomise a state, which then translates into a set of values for each element in the multivariate distribution. If the finite population is easily countable this procedure is simply equivalent to randomly selecting a unit in the population, as illustrated in the following example:

Example 3.4. To be added...

For larger populations, the same idea can be implemented by enumerating the possible values of all units in the population; each possible set of values corresponds to one state. The probability of each state is then computed, which gives a distribution function $F_Y(x)$ for the states. The state can then be randomised using the inverse transform method, and the values of the random variables are then set according to the state.

Example 3.5. To be added...

It can be noticed that the order in which units or states are enumerated might have consequences for the application of variances reduction techniques, especially complementary random numbers (cf. section 5.1.2).

To be added: Discussion of how continuous distributions can be approximated by a finite population to allow the above described method to be applied (rather than the general method below)...

General Method

The inverse transform method is only applicable to univariate distributions. The idea of the general method is to generate one value at a time for a multivariate distribution. This requires that a conditional probability based on the outcome outcome of the already generated random values is computed, and then the inverse transform method to this conditional distribution. The method works on any multivariate distribution, but the disadvantage is that the computation of the conditional probability distributions can be quite time-consuming for complex distribution.

Consider a multivariate distribution of $Y = [Y_1, \dots, Y_K]$ with the density function f_Y . We can then apply the following algorithm:

- **Step 1.** Calculate the density function of the first element, f_{Y_1} .
- **Step 2.** Generate a value of the first element according to f_{Y_1} .
- **Step 3.** Calculate the conditional density function of the next element, i.e.,

$$f_{Y_k|[Y_1, \dots, Y_{k-1}]} = [\psi_1, \dots, \psi_{k-1}]'$$

- **Step 4.** Randomise the value of the k :th element according to $f_{Y_k|[Y_1, \dots, Y_{k-1}]} = [\psi_1, \dots, \psi_{k-1}]'$
- **Step 5.** Repeat step 3–4 until all elements have been randomised.

Example 3.6. To be added...

Approximative Method

To be added...

Exercises

3.1 Figure 3.2 shows the distribution function of the continuous random variable Y . Apply the inverse transform method to generate a value of Y based on the following random numbers from a $U(0, 1)$ -distribution:

- a) 0.16.
- b) 0.28.
- c) 0.55.
- d) 0.96.

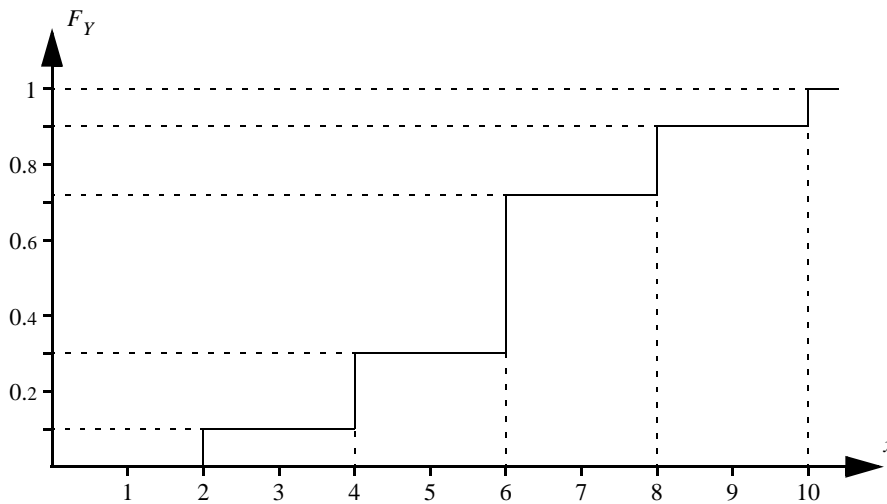


Figure 3.1 Distribution function in exercise 3.2.

3.2 Figure 3.1 shows the distribution function of the continuous random variable Y . Apply the inverse transform method to generate a value of Y based on the following random numbers from a $U(0, 1)$ -distribution:

- a) 0.10.
- b) 0.25.
- c) 0.63.
- d) 0.91.

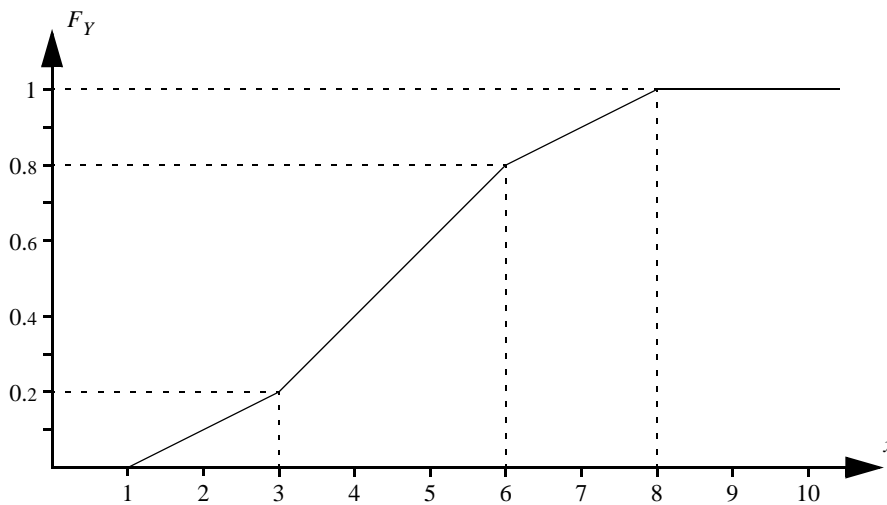


Figure 3.2 Distribution function in exercise 3.1.

3.3 A system has two binary inputs, Y_1 and Y_2 , such that $P(Y_1 = 0) = 0.87$, $P(Y_1 = 1) = 0.13$, $P(Y_2 = 0) = 0.94$ and $P(Y_2 = 1) = 0.06$. The probability distributions of these two inputs have been combined into a

single joint distribution as shown in figure 3.3. Apply the inverse transform method to convert the random value 0.46 from a $U(0, 1)$ -distribution into values of Y_1 and Y_2 .

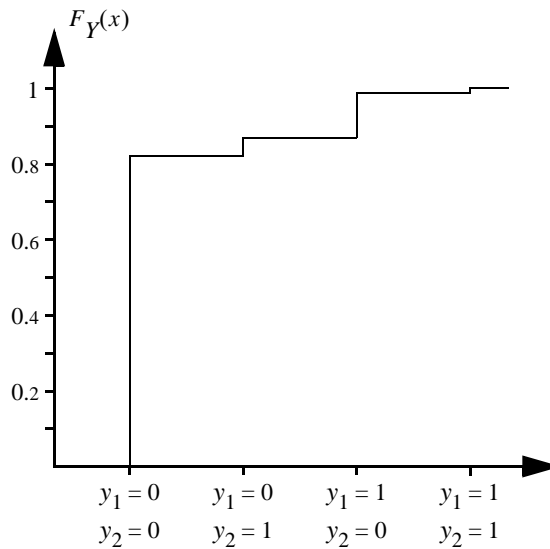


Figure 3.3 Distribution function in exercise 3.3.

3.4 Y_1 and Y_2 are two correlated random variables. The joint frequency function is given by

$$f_Y(x_1, x_2) = \begin{cases} 0.3 & \text{if } x_1 = 1, x_2 = 1, \\ 0.2 & \text{if } x_1 = 1, x_2 = 2, \\ 0.2 & \text{if } x_1 = 2, x_2 = 1, \\ 0.3 & \text{if } x_1 = 2, x_2 = 2, \\ 0 & \text{all other } (x_1, x_2). \end{cases}$$

Generate random values of Y_1 and Y_2 . You may use any of the $U(0, 1)$ -distributed random numbers provided in table 3.4.

Table 3.4 List of random values for exercise 3.4.

| Number | 1 | 2 | 3 | 4 |
|--------|------|------|------|------|
| Value | 0.82 | 0.13 | 0.63 | 0.28 |

3.5 Y_1 and Y_2 are two correlated random variables. The density function of Y_1 is given by

$$f_{Y_1}(x) = \begin{cases} 0 & \text{if } x < 0, \\ 0.05 & \text{if } 0 \leq x \leq 5, \\ 0.15 & \text{if } 5 < x \leq 10, \\ 0 & \text{if } 10 \leq x. \end{cases}$$

For $Y_1 \leq 5$, the conditional density function of Y_2 is given by

$$f_{Y_2}(x | Y_1 \leq 5) = \begin{cases} 0 & \text{if } x < 0, \\ 0.08 & \text{if } 0 \leq x \leq 5, \\ 0.12 & \text{if } 5 < x \leq 10, \\ 0 & \text{if } 10 \leq x, \end{cases}$$

whereas for $Y_1 > 5$ we have

$$f_{Y_2}(x | Y_1 > 5) = \begin{cases} 0 & \text{if } x < 0, \\ 0.04 & \text{if } 0 \leq x \leq 5, \\ 0.16 & \text{if } 5 < x \leq 10, \\ 0 & \text{if } 10 \leq x. \end{cases}$$

Generate random values of Y_1 and Y_2 using the random numbers 0.22 and 0.52 from a $U(0, 1)$ -distribution.

3.6 Y_1 and Y_2 are two correlated random variables. The joint density function is given by

$$f_{Y_1, Y_2}(x_1, x_2) = \begin{cases} 0.008 & \text{if } 0 \leq x_1 \leq 5, 0 \leq x_2 \leq 10, \\ 0.008 & \text{if } 5 \leq x_1 \leq 10, 0 \leq x_2 \leq 5, \\ 0.016 & \text{if } 5 \leq x_1 \leq 10, 5 < x_2 \leq 10, \\ 0 & \text{all other } (x_1, x_2). \end{cases}$$

Generate random values of Y_1 and Y_2 using the random numbers 0.22 and 0.64 from a $U(0, 1)$ -distribution.

SIMPLE SAMPLING

The most straightforward approach to Monte Carlo simulation is to collect samples completely at random. This sampling strategy is referred to as *simple sampling* and a Monte Carlo simulation based on simple sampling is sometimes referred to as a *crude Monte Carlo simulation*. Although we will see in the next chapter that it is in fact possible to get better results using other sampling methods, it is necessary to understand simple sampling before other methods are studied. Moreover, in some cases simple sampling will be the most efficient simulation method, either due to the properties of the study or because the time to develop a more advanced simulation method is longer than the time to obtain a reasonable result from simple sampling.

4.1 Estimating Expectation Values

The idea of a Monte Carlo simulation is to collect random samples and based on these observations try to compute one or more values, for example a statistical property of a random variable or a deterministic value such as the mathematical constant π in Buffon's needle experiment. However, as the result is depending on random observations, it is inevitable that we introduce a random error—the result of a Monte Carlo simulation will most likely not be exactly equal to the true value. In that sense, the result of a Monte Carlo simulation is just an estimate, which we would like to be as close as possible to the true value.

It is important to notice that since the result of a Monte Carlo simulation is a function of several random samples, the result must also be a random variable. This is reflected in the notation for estimates that will be used in this compendium. If we are discussing general properties of an estimate calculated using one or another simulation method, we will denote the estimate by an upper-case Latin letter corresponding to the statistical property that we are trying to estimate. For example, the expectation value of the random variable X is denoted μ_X , which means that we choose the symbol M_X for the random variable representing the estimate of μ_X . However, if we are actually computing an estimate then we are in fact studying an outcome of M_X , and we will then use the symbol m_X , i.e., the lower-case Latin counterpart of μ_X .

Accuracy and Precision

In everyday language, we may use the word “accuracy” to describe if an estimate is close or not to the true value. However, in science, accuracy should be separated from “precision”. In this section, the meaning and difference between these two notions will be described, and we will at the same time discuss desirable properties of an estimate.

What we want from an estimate is that it should be as close as possible to the true value. Each simulation method that we create will generate estimates from a certain probability distribution; therefore, we may study this probability distribution of the estimate, in order to determine if it is likely or not that we will get a result close to the true value. It can then be seen that there are two different ways that an estimate can be “close” to the true value.

Let us start by examining the difference between the expectation value of the estimate, $E[M_X]$, and the

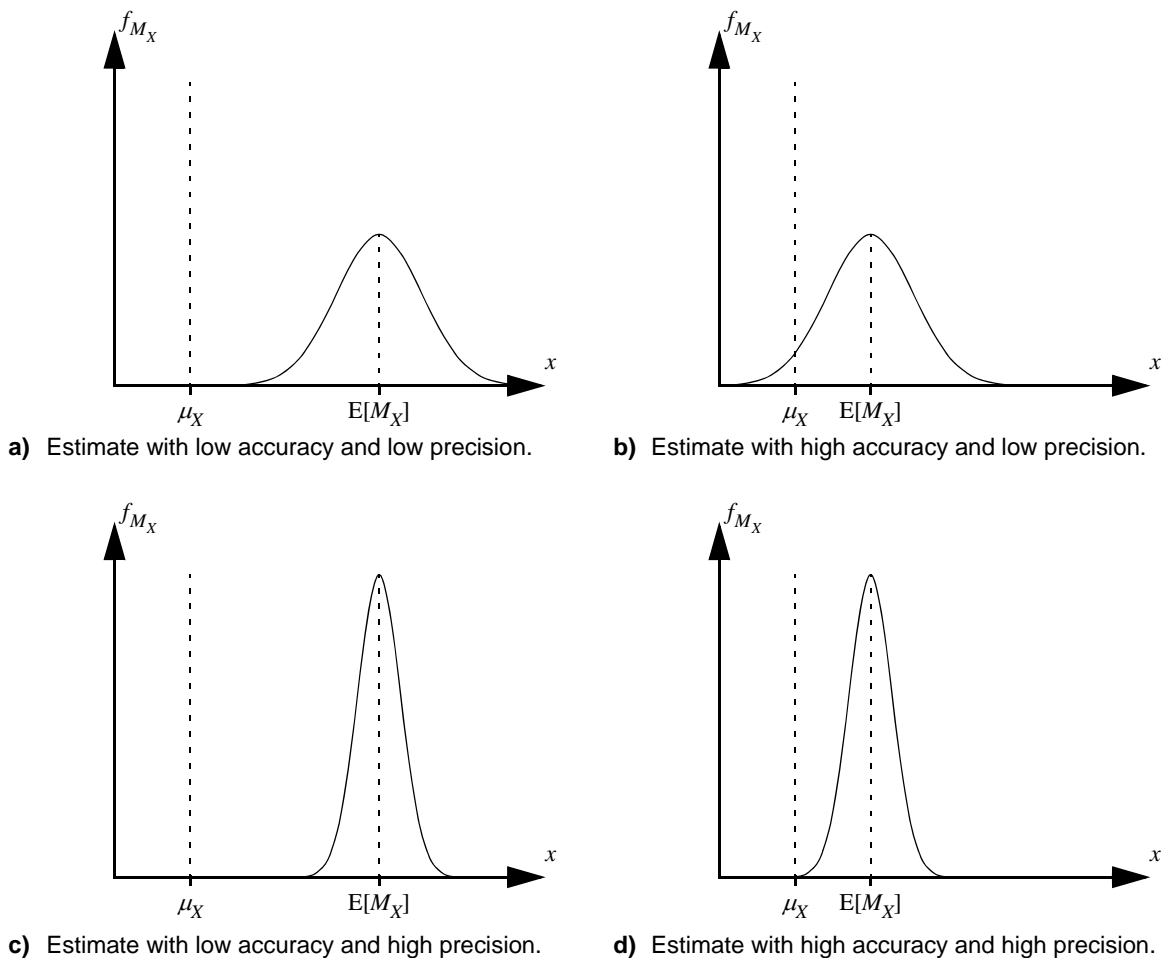


Figure 4.1 Illustration of the meaning of “accuracy” and “precision” respectively.

true value, μ_X . This difference is referred to as the *accuracy* of the estimate. Examples of estimates with low and high accuracy are shown in figures 4.1a and 4.1b. We can see that a high accuracy means that the difference between $E[M_X]$ and μ_X is small; consequently, we have a better probability that the resulting estimate from a simulation is close to the true value compared to a simulation with lower accuracy. Since the accuracy is a constant for a given simulation method (remember that each simulation method will result in its own probability distribution of the estimate) it represents the *systematical error* in a simulation. The ideal is of course if we can set up a simulation with no systematical error, i.e., that $E[M_X] = \mu_X$. In this case, we say that we have an *unbiased estimate*.

Now, we may continue by examining the variance of the estimate, $\text{Var}[M_X]$. If we go back to the definition, the variance is the expected quadratic deviation from the expectation value; hence, the variance describes how much a random variable will vary around its expectation value. A low variance means that it is likely that we only have a small deviation between a single observation of the random variable and its expectation value, whereas a high variance means that it is more likely to have large deviations. The variance of the estimate is referred to as the *precision* of the estimate. The precision is inversely proportional to the variance, i.e., a large variance of the estimate means that we have a low precision, whereas a low variance yields a high precision (cf. figures 4.1c and 4.1d). The precision represents the random error of a Monte Carlo simulation. Unlike systematical errors, this is something that cannot be avoided when sampling. What we would like to is therefore to keep the random error as small as possible.

The Law of Large Numbers

To be added: Introduction...

Theorem 4.1. (Law of Large Numbers): If x_1, \dots, x_n are independent observations of the random variable X then

$$m_X = \frac{1}{n} \sum_{i=1}^n x_i$$

is an unbiased estimate of $E[X]$.

Proof: M_X is an unbiased estimate of $E[X]$ if $E[M_X] = E[X]$. Let T_i denote the number of times that unit i appears in the samples; hence, the outcome of T_i is an integer between 0 and n . The estimate of the expectation value can then be expressed as

$$M_X = \frac{1}{n} \sum_{i=1}^N T_i \mathcal{X}_i.$$

The number of successful trials when n trials are performed and the probability of success is p in each trial is Binomial-distributed, i.e. T_i is $B(n, 1/N)$ -distributed.

$$E[M_X] = E\left[\frac{1}{n} \sum_{i=1}^N T_i \mathcal{X}_i\right] = \sum_{i=1}^N \frac{1}{n} E[T_i] \mathcal{X}_i = \{\text{the expectation value of a } B(n, p)\text{-distribution is } n \cdot p\} = \sum_{i=1}^N \frac{1}{n} \mathcal{X}_i = \frac{1}{N} \sum_{i=1}^N \mathcal{X}_i = E[X]. \blacksquare$$

Notice the similarity between the definition of expectation value (definition 2.11) and the formula for the estimate of the expectation value (theorem 4.1 above):

$$\mu_X = \frac{1}{N} \sum_{i=1}^N \mathcal{X}_i \leftrightarrow m_X = \frac{1}{n} \sum_{i=1}^n x_i. \tag{4.1}$$

When calculating the expectation value analytically, we enumerate all units in the population and compute the mean value, whereas in simple sampling we enumerate all selected samples and compute the mean value.

Replacement

To be added: Definition of replacement...

To be added: Discussion of the precision of the estimated expectation value...

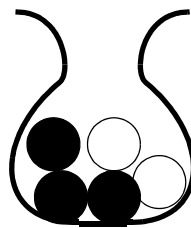


Figure 4.2 Simple urn experiment.

Example 4.1. Consider an urn with five balls. The balls are either white or black. (They also have different sizes, but that we will ignore in this example.) Introduce a random variable X representing the colour of a randomly chosen ball from the urn, such that

$$X = \begin{cases} 0 & \text{if the ball is white,} \\ 1 & \text{if the ball is black.} \end{cases}$$

Examine the probability distribution of M_X if five samples are collected with and without replacement respectively.

Solution: To be added: Define frequency function, compute μ_X , explain figure 4.3 (and add the probability of each state during the sample process)...

Theorem 4.2. In simple sampling, the variance of the estimated expectation value is

$$\text{Var}[M_X] = \frac{\text{Var}[X]}{n} \cdot \frac{N-n}{N}.$$

Proof: To be added?

The factor $(N-n)/N$ is called fpc (*finite population correction*). For infinite populations we get

$$\text{Var}[M_X] = \frac{\text{Var}[X]}{n}. \tag{4.2}$$

To be added: Interpretation of the theorem & examples...

4.2 Other Estimates

To be added: Introduction...

Estimating Variance

To be added: Introduction...

From the comparison in (4.1) one might guess that an estimate of the variance can be obtained by

$$\frac{1}{n} \sum_{i=1}^n (x_i - \mu_X)^2. \tag{4.3}$$

However, this estimate requires that we know the true expectation value, and if this was the case we would also be able to analytically compute the variance. Hence, (4.3) cannot be applied in practice. But what about replacing the true expectation value by the estimated expectation value, i.e.,

$$\frac{1}{n} \sum_{i=1}^n (x_i - m_X)^2. \tag{4.4}$$

It turns out that the (4.4) is indeed an estimate of $\text{Var}[X]$, although, the estimate is biased.

To be added: Explanation why (4.4) is biased?

Theorem 4.3. If x_1, \dots, x_n are independent observations of the random variable X then

$$s_X^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - m_X)^2$$

is an unbiased estimate of $\text{Var}[X]$.

The proof of theorem 4.3 is not trivial and will therefore will be omitted from this presentation. As a substitute, we may verify the theorem as well as (4.3) and (4.4) by an example:

Example 4.2. Consider the urn experiment from example 4.1. Examine the probability distribution of the estimates (4.3), (4.4) and theorem 4.3 respectively if five samples are collected (with replacement).

Solution: The true variance of the urn experiment is given by

$$\text{Var}[X] = \sum_x f_X(x)(x - \mu_X)^2 = 0.4 \cdot (0 - 0.6)^2 + 0.6 \cdot (1 - 0.6)^2 = 0.24.$$

As the order in which the samples appear is of no interest for neither the estimated mean nor the estimated variance, we can distinguish between six possible outcomes, as listed in table 4.1 below. The estimates of mean and variance are then computed for each combination of collected samples. The probability for a certain result can be computed using the

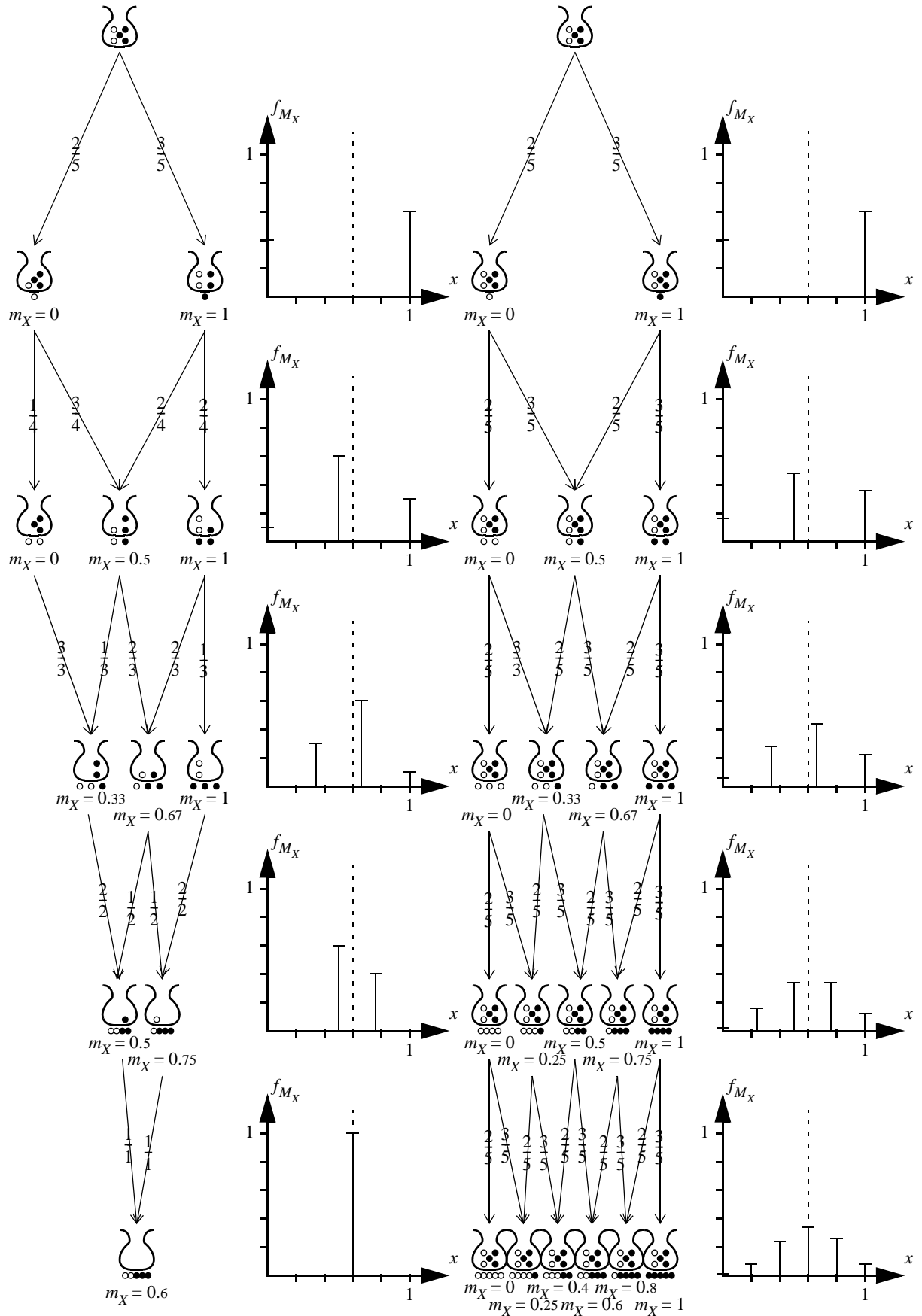


Figure 4.3 Probability distribution of the estimated expectation value in the urn experiment. Sampling without replacement is shown to the left and sampling with replacement is shown to the right.

event tree in figure 4.3.

Table 4.1 Probability distribution of the estimated variance in the urn experiment.

| Collected samples | ○○○○ | ○○○● | ○○●● | ○○●● | ○○●● | ●●●● |
|--|---------------------|----------------------|----------------------|-------------------------|----------------------|----------------------|
| Probability | $\frac{32}{3\,125}$ | $\frac{240}{3\,125}$ | $\frac{720}{3\,125}$ | $\frac{1\,080}{3\,125}$ | $\frac{810}{3\,125}$ | $\frac{243}{3\,125}$ |
| Estimated mean, m_X | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1 |
| Estimated variance | | | | | | |
| $\frac{1}{n} \sum_{i=1}^n (x_i - \mu_X)^2$ | 0.36 | 0.32 | 0.28 | 0.24 | 0.2 | 0.16 |
| $\frac{1}{n} \sum_{i=1}^n (x_i - m_X)^2$ | 0 | 0.16 | 0.24 | 0.24 | 0.16 | 0 |
| $\frac{1}{n-1} \sum_{i=1}^n (x_i - m_X)^2$ | 0 | 0.2 | 0.3 | 0.3 | 0.2 | 0 |

From the results in table 4.1 we can compute the expectation value of the estimated variance for the three different methods:

$$E\left[\frac{1}{n} \sum_{i=1}^n (x_i - \mu_X)^2\right] = \frac{32}{3\,125}0.36 + \frac{240}{3\,125}0.32 + \frac{720}{3\,125}0.28 + \frac{1\,080}{3\,125}0.24 + \frac{810}{3\,125}0.2 + \frac{243}{3\,125}0.16 = 0.24.$$

$$E\left[\frac{1}{n} \sum_{i=1}^n (x_i - m_X)^2\right] = \frac{32}{3\,125}0 + \frac{240}{3\,125}0.16 + \frac{720}{3\,125}0.24 + \frac{1\,080}{3\,125}0.24 + \frac{810}{3\,125}0.16 + \frac{243}{3\,125}0 = 0.192.$$

$$E\left[\frac{1}{n-1} \sum_{i=1}^n (x_i - m_X)^2\right] = \frac{32}{3\,125}0 + \frac{240}{3\,125}0.2 + \frac{720}{3\,125}0.2 + \frac{1\,080}{3\,125}0.3 + \frac{810}{3\,125}0.3 + \frac{243}{3\,125}0 = 0.24.$$

These results confirm that (4.3) and theorem 4.3 are unbiased estimates (because the expectation value of the estimate is equal to the true value), whereas (4.4) tends to underestimate the variance.

To be added: Show how S_X^2 can be computed using sums of x_i and x_i^2 .

To be added: Estimating covariance...

Estimating Probability Distributions

To be added...

Estimating Covariance

To be added...

4.3 Application

To be added: Introduction...

Stopping Rules

To be added: Discussion of how many samples should be analysed in a Monte Carlo simulation...

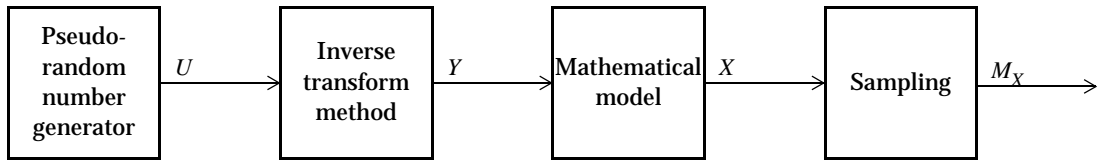


Figure 4.4 Principle of simple sampling.

Definition 4.4. The coefficient of variation for the estimate M_X is given by

$$A_X = \frac{\sqrt{\text{Var}[M_X]}}{M_X}.$$

Confidence Intervals

To be added: Confidence intervals...

Definition 4.5. A confidence interval is an interval which has some specific probability (confidence level) to include the true value.

Theorem 4.6. Let t_α be the value such that $\Phi(t_\alpha) = 1 - \alpha/2$.¹ If M_X is normally distributed then $m_X \pm t_\alpha \cdot s_X / \sqrt{n}$ is a confidence interval with confidence level $1 - \alpha$.

Table 4.2 Typical values of t_α

| | | | |
|--------------------------------|--------|--------|--------|
| Confidence level, $1 - \alpha$ | 95% | 99% | 99.9% |
| t_α | 1.9600 | 2.5758 | 3.2905 |

Simulation Procedure

To be added: Discussion...

Example 4.3 (simple sampling of Akabuga District). To be added...

Exercises

4.1 The following results have been obtained from a Monte Carlo simulation using simple sampling:

$$\sum_{i=1}^{500} x_i = 248, \quad \sum_{i=1}^{500} x_i^2 = 252.$$

- a) Calculate the estimated expectation value of X .
- b) Calculate the estimated variance of X .
- c) Calculate the coefficient of variation.
- d) Assume that the coefficient of variation is used in the stopping criteria, and that the relative tolerance level is set to 0.01. Should more samples be generated or can the simulation be stopped at this point?
- e) Calculate a 95% confidence interval for the expectation value of X . It can be assumed that M_X is normally distributed.

1. $\Phi(x)$ is the distribution function of a $N(0, 1)$ -distributed random variable.

4.2 The following results have been obtained from a Monte Carlo simulation using simple sampling:

$$\sum_{i=1}^{100} x_i = 20\,000, \quad \sum_{i=1}^{100} x_i^2 = 4\,000\,000.$$

- a) Calculate the estimated expectation value of X .
 - b) Calculate the estimated variance of X .
 - c) Would it be appropriate to have a stopping rule for the simulation of X , which stops the simulation if the coefficient of variation is less than 0.01? Do not forget to motivate your answer!
- 4.3** Assume that the objective of a Monte Carlo simulation is to find an interval $m_X \pm 0.1m_X$, which is a 95% confidence interval for the expectation value $E[X]$. The stopping criteria for the simulation is to test if the coefficient of variation is less than the relative tolerance level, ρ . Calculate an appropriate value for ρ .

VARIANCE REDUCTION TECHNIQUES

It was shown in the previous chapter that the variance of an estimated expectation value, $Var[M_X]$, is related to the precision of the simulation; a low variance means that it is more likely that the result will be accurate, whereas a high variance means that there is a larger risk that the result will be inaccurate. We also learned that $Var[M_X]$ is depending on the probability distribution of the samples variable (which we cannot affect) and the number of samples (which we do control). However, until now we have studied simple sampling, where the samples are selected completely at random (i.e., each unit in the population has the same probability of being selected). Interestingly, if we manipulate the selection of samples, the variance of the estimate can be lower than for simple sampling. Such methods to improve the precision of a Monte Carlo simulation are referred to as *variance reduction techniques*. This chapter will describe six variance reduction techniques.

5.1 Complementary Random Numbers

The idea behind complementary random numbers is to reduce the influence of random fluctuations, which always appear in sampling, by creating a negative correlation between samples. In practice, this means that the generation of random numbers is manipulated in such a way that the probability of an even spread over the whole population increases.

5.1.1 Principle

Assume that the expectation value $E[X] = \mu_X$ has been estimated in two separate simulations, i.e., we have two estimates M_{X1} and M_{X2} such that

$$E[M_{X1}] = E[M_{X2}] = \mu_X. \quad (5.1)$$

It is not surprising that the mean of the two estimates is also an estimate of μ_X , and we can easily verify that this is the case, because

$$E\left[\frac{M_{X1} + M_{X2}}{2}\right] = \frac{1}{2}(E[M_{X1}] + E[M_{X2}]) = \frac{1}{2}(\mu_X + \mu_X) = \mu_X. \quad (5.2)$$

As the expectation value of the mean estimate is equal to μ_X , the mean estimate is in itself an unbiased estimate of μ_X in accordance to the discussion in section 4.1. The interesting question is now how precise this mean estimate is compared to simple sampling with the same total number of samples. From (4.2) we have that simple sampling with in total n samples results in

$$Var[M_X] = \frac{Var[X]}{n} = \frac{\sigma_X^2}{n}. \quad (5.3)$$

Now assume that the two estimates M_{X1} and M_{X2} each include $n/2$ samples (which means that the total number of samples is still n) then variance of the mean estimate is given by

$$\text{Var}\left[\frac{M_{X1} + M_{X2}}{2}\right] = \frac{1}{4}\text{Var}[M_{X1}] + \frac{1}{4}\text{Var}[M_{X2}] + \frac{1}{4} \cdot 2\text{Cov}(M_{X1}, M_{X2}). \quad (5.4)$$

If M_{X1} and M_{X2} both are independent estimates obtained with simple sampling then we get $\text{Var}[M_{X1}] = \text{Var}[M_{X2}] = \sigma_X^2$ and $\text{Cov}[M_{X1}, M_{X2}] = 0$. Hence, (5.4) yields

$$\text{Var}\left[\frac{M_{X1} + M_{X2}}{2}\right] = \frac{1}{4}(\text{Var}[M_{X1}] + \text{Var}[M_{X1}]) = \frac{\text{Var}[M_{X1}]}{2} = \frac{\sigma_X^2}{n}. \quad (5.5)$$

By comparing (5.3) and (5.5) we see that the precision is the same, which is also what we should expect—running one simulation of n independent samples should be the same thing as combining the results of two independent simulations of $n/2$ independent samples each.

However, the interesting part is that if the two estimates are *not* independent but negatively correlated then (5.4) will result in a variance that is lower than the variance for simple sampling with the same amount of samples. The question is then how we should proceed in order to find estimates that are negatively correlated. A straightforward method is to use *complementary random numbers* when we generate scenarios for a Monte Carlo simulation.

- We start with a random number from a random number generator, which we have seen in section 3.1, corresponds to a $U(0, 1)$ -distributed random variable. If U is a value from the random number generator then we define $U^* = 1 - U$ as the complementary random number of U . It is easy to verify that U and U^* are negatively correlated with $\rho_{U, U^*} = -1$.
- Then we transform both the original random number and the complementary random number into the probability distribution of the input. If this is done using the inverse transform method then we get that $Y = F_Y^{-1}(U)$ and $Y^* = F_Y^{-1}(U^*)$. These values will also be negatively correlated, but the transformation may weaken the correlation, i.e., we get $\rho_{Y, Y^*} \geq -1$. This also holds for the other transformation methods presented in section 3.2.
- Next we compute the value of the output for both original and the complementary input value, i.e., $X = g(Y)$ and $X^* = g(Y^*)$. If the simulated system is such that there is a correlation between the input and output values then the original and complementary output values will also be negatively correlated, but again the correlation might be weakened, i.e., we get $\rho_{X, X^*} \geq \rho_{Y, Y^*}$.
- Finally, we let M_{X1} be an estimate based on n samples of X , whereas M_{X2} is obtained from sampling the corresponding values of X^* . Obviously, we will now have two estimates that are negatively correlated.

5.1.2 Application

To be added: Introduction...

Multiple Inputs

To be added: Discussion on how to manage systems with multiple inputs...

Auxiliary Inputs

Complementary random numbers result in a variance reduction if there is a negative correlation between the original and complementary result variables (X and X^*). The challenge when setting up a simulation is however that we cannot directly control the values of the result variables, but only the values of the inputs (Y and Y^*)—complementary random numbers will be efficient if the negative correlation between Y and Y^* also results in a negative correlation between $X = g(Y)$ and $X^* = g(Y^*)$. This may not be true for all inputs to a system, and we should then not apply complementary random numbers to these inputs.

Sometimes it might even not be possible to find any input variable that fulfils the criterion above. This does however not necessarily mean that complementary random numbers are not applicable to this system, as there might be a possibility to introduce an auxiliary input that does fulfil the criterion. An auxiliary input, Y_E , is defined as an arbitrary function of the actual inputs to the system, i.e.,

$$Y_E = h(Y_1, \dots, Y_J). \quad (5.6)$$

A straightforward choice of the arbitrary function h would be to use the sum of some of the inputs.

Example 5.1 (auxiliary inputs for a small multi-area power system). Consider a power system divided in three areas. The inputs to a simulation of this system is the available generation capacity in each unit, $\bar{G}_1, \dots, \bar{G}_4$ and the load in each area, D_1, D_2, D_3 . The objective of the simulation is to study the total operation cost and the reliability of supply. Assume that complementary random numbers should be used to improve the efficiency of the simulation. To which inputs should we apply complementary random numbers?

Solution: Each individual input will only have a weak impact on the outputs. For example, if one unit is unavailable we cannot predict that the system will have a higher operation cost or that there will be load shedding, because the other units can be available or the load can be low. However, we can introduce two auxiliary inputs, namely the total available generation capacity, $\bar{G}^{\text{tot}} = \bar{G}_1 + \bar{G}_2 + \bar{G}_3 + \bar{G}_4$, and total demand, $D^{\text{tot}} = D_1 + D_2 + D_3$. A negative correlation between two values of the auxiliary inputs will also result in a negative correlation between at least one of the result variables, as illustrated in table 5.1.

Table 5.1 Correlations between auxiliary inputs and result variables in example 5.1.

| Total available generation capacity, \bar{G}^{tot} | Total demand, D^{tot} | Total operation cost, TOC | Loss of load, $LOLO$ |
|---|--------------------------------|-----------------------------|----------------------|
| Low | Low | Low | Not likely |
| High | Low | Low | Very unlikely |
| Low | High | High | Quite likely |
| High | High | High | Not likely |

In order to correctly simulate a system considering auxiliary inputs, we need to have the probability distribution of the auxiliary inputs. Although this may require some computational efforts, it should not be a problem as the probability distribution of all the original inputs is known. Moreover, we must make sure that the random values of the auxiliary inputs are consistent with the random values of the original inputs. As the auxiliary inputs are a function of some original inputs, there will clearly be a correlation between auxiliary and original inputs. We could therefore identify their joint probability distribution and randomise values according to this distribution.

To be added: Further discussion (challenge of generating correlated numbers, cf. section 3.2.2).

Example 5.2 (joint probability distribution of available generation capacity). Data for the generating units in example 5.1 are shown in table 5.2. Identify the joint probability distribution for the total generation capacity and the generation capacity in each unit. Moreover, generate a scenario using the value 0.17 from a $U(0, 1)$ -distribution as well as a complementary scenario.

Table 5.2 Data for the power system in example 5.2.

| Unit, g | Area, n | Installed capacity, \bar{G}_g [MW] | Availability [%] |
|-----------|-----------|--------------------------------------|------------------|
| 1 | 1 | 500 | 95 |
| 2 | 2 | 500 | 95 |
| 3 | 2 | 800 | 90 |
| 4 | 3 | 1 000 | 90 |

Solution: As each unit can either be available or unavailable, we will in total have sixteen possible states for the available generation capacity of all units. For each of these states we can compute the total available generation capacity and the probability of that state. The result of this enumeration, sorted according to increasing values of \bar{G}^{tot} are listed in table 5.3.

To be added: Comments... (for example, why is there a difference between state 2 and 3)

To be added: Figure showing the distribution function and the result of applying the

Table 5.3 Joint probability distribution of the available generation capacity in example 5.2.

| State, i | Total available generation capacity, \bar{G}^{tot} [MW] | Available generation capacity in unit 1, \bar{G}_1 [MW] | Available generation capacity in unit 2, \bar{G}_2 [MW] | Available generation capacity in unit 3, \bar{G}_3 [MW] | Available generation capacity in unit 4, \bar{G}_4 [MW] | Probability, $f_{\bar{G}}$ |
|------------|--|---|---|---|---|--|
| 1 | 0 | 0 | 0 | 0 | 0 | $0.05 \cdot 0.05 \cdot 0.1 \cdot 0.1 = 0.000025$ |
| 2 | 500 | 500 | 0 | 0 | 0 | $0.95 \cdot 0.05 \cdot 0.1 \cdot 0.1 = 0.000475$ |
| 3 | 500 | 0 | 500 | 0 | 0 | $0.05 \cdot 0.95 \cdot 0.1 \cdot 0.1 = 0.000475$ |
| 4 | 800 | 0 | 0 | 800 | 0 | $0.05 \cdot 0.05 \cdot 0.9 \cdot 0.1 = 0.000225$ |
| 5 | 1 000 | 0 | 0 | 0 | 1 000 | $0.05 \cdot 0.05 \cdot 0.1 \cdot 0.9 = 0.000225$ |
| 6 | 1 000 | 500 | 500 | 0 | 0 | $0.95 \cdot 0.95 \cdot 0.1 \cdot 0.1 = 0.009000$ |
| 7 | 1 300 | 500 | 0 | 800 | 0 | $0.95 \cdot 0.05 \cdot 0.9 \cdot 0.1 = 0.004275$ |
| 8 | 1 300 | 0 | 500 | 800 | 0 | $0.05 \cdot 0.95 \cdot 0.9 \cdot 0.1 = 0.004275$ |
| 9 | 1 500 | 500 | 0 | 0 | 1 000 | $0.95 \cdot 0.05 \cdot 0.1 \cdot 0.9 = 0.004275$ |
| 10 | 1 500 | 0 | 500 | 0 | 1 000 | $0.05 \cdot 0.95 \cdot 0.1 \cdot 0.9 = 0.004275$ |
| 11 | 1 800 | 0 | 0 | 800 | 1 000 | $0.05 \cdot 0.05 \cdot 0.9 \cdot 0.9 = 0.002025$ |
| 12 | 1 800 | 500 | 500 | 800 | 0 | $0.95 \cdot 0.95 \cdot 0.9 \cdot 0.1 = 0.081225$ |
| 13 | 2 000 | 500 | 500 | 0 | 1 000 | $0.95 \cdot 0.95 \cdot 0.1 \cdot 0.9 = 0.081225$ |
| 14 | 2 300 | 500 | 0 | 800 | 1 000 | $0.95 \cdot 0.05 \cdot 0.9 \cdot 0.9 = 0.038475$ |
| 15 | 2 300 | 0 | 500 | 800 | 1 000 | $0.05 \cdot 0.95 \cdot 0.9 \cdot 0.9 = 0.038475$ |
| 16 | 2 800 | 500 | 500 | 800 | 1 000 | $0.95 \cdot 0.95 \cdot 0.9 \cdot 0.9 = 0.731025$ |

inverse transform method using $U = 0,17$ and $U^* = 0,83$.

Complementary Random Numbers for Finite Correlated Multivariate Distributions

To be added: Examples showing how states can be ordered to facilitate complementary random numbers...

Simulation Procedure

To be added: Overview, equations, block diagram and example...

5.2 Dagger Sampling

This variance reduction technique is based on a similar principle as complementary random numbers. Dagger sampling is however limited to two-state probability distributions, where complementary random numbers are not very effective.

To be added: Further comments?

5.2.1 Principle

Consider a two-state random variable Y with the frequency function

$$f_Y(x) = \begin{cases} 1-p & \text{if } x = a, \\ p & \text{if } x = b, \\ 0 & \text{otherwise,} \end{cases} \tag{5.7}$$

where $p < 0.5$. This probability distribution is clearly fulfilling the criteria of a duogeneous population (cf. definition 2.6), with the value a being the conformist units and b the diverging units.

In dagger sampling, random values of Y are not generated by the inverse transform method, but using a dagger transform:

Theorem 5.1. (Dagger Transform) If U is a $U(0, 1)$ -distributed random number then Y is a

distributed according to the frequency function (5.7) if Y is calculated according to

$$F_{Y_j^\ddagger}(x) = \begin{cases} b & \text{if } (j-1)p \leq x < jp, \\ a & \text{otherwise,} \end{cases} \quad \text{for } j = 1, \dots, S,$$

where S is the largest integer such that $S \leq 1/p$.

The value S in theorem 5.1 is referred to as the dagger cycle length. It should be noted that one random number from the pseudorandom number generator is used to generate S values of Y .

Example 5.3. To be added...

We have already seen in section 5.1.1 that a negative correlation between input values can result in a variance reduction compared to simple sampling. We can also easily see that there is a negative correlation between the random values in a dagger cycle, because the diverging unit can never appear more than once in a dagger cycle. This means that if we know that the j :th value was equal to the diverging unit then we know that all the other $S - 1$ values are equal to the conformist unit; hence, the j :th value and the other values are varying in opposite directions, which is characteristic for a negative correlation.

We can also verify the negative correlation by going back to the mathematical definition. Let us start by investigating the product of two values in a dagger cycle, i.e., $Y_j Y_k$. There are only two possible values of this product, aa or ab , since the diverging unit cannot appear more than once in the dagger cycle. We can also observe that for all dagger transforms there will only be two intervals there either the j :th or the k :th value of dagger cycle are equal to the diverging unit b ; hence, the probability for this result is $2p$. This means that the expectation value of the product is given by

$$E[Y_j Y_k] = 2pab + (1 - 2p)aa. \quad (5.8)$$

Moreover, we can compute the expectation value of each value as

$$E[Y_j] = E[Y_k] = (1 - p)a + pb. \quad (5.9)$$

The covariance between Y_j and Y_k can now be computed according to definition 2.14:

$$\text{Cov}[Y_j, Y_k] = E[Y_j Y_k] - E[Y_j]E[Y_k] = pab + (1 - 2p)aa - ((1 - p)a + pb)^2 = -p^2(a + b)^2 < 0. \quad (5.10)$$

To be added: Example with figure...

5.2.2 Application

To be added: Introduction...

Multiple Inputs

To be added: Discussion on how to manage inputs with different dagger cycle lengths...

Simulation Procedure

To be added: Overview, equations, block diagram and example...

5.3 Control Variates

To be added: Introduction...

5.3.1 Principle

5.3.2 Application

To be added: Introduction...

Finding a Simplified Model

To be added: Discussion on how to create suitable simplified models...

Simulation Procedure

To be added: Impact on coefficient of variation depending on how the control variate method is implemented...

To be added: Overview, equations, block diagram and example...

5.4 Correlated Sampling

To be added: Introduction...

5.4.1 Principle

5.4.2 Application

To be added: Introduction...

To be added: Overview, equations, block diagram and example...

5.5 Importance Sampling

The aim of importance sampling is to focus the attention of the simulation to those scenarios that are of high importance for the final result. A typical example is a duogeneous population, where it is necessary to find the rare diverging units in order to correctly estimate the expectation value of the population. In importance sampling, the selection of units (i.e., the randomisation of scenarios) is manipulated to increase the probability of finding interesting scenarios. Thus, in contrast to simple sampling, where all units have the same probability of being selected and there all observations have the same weight, importance sampling results in some units having a higher chance of being selected, but on the other hand these units have a lesser weight in the final result.

5.5.1 Principle

Assume that we have a population \mathcal{Y} , for which we want to compute the expectation value $E[X] = E[g(Y)]$. However, instead of selecting units from \mathcal{Y} and evaluate using the model $g(Y)$, we choose to select unit from another population \mathcal{Z} and then we sample

$$X(Z) = w(Z) \cdot g(Z), \tag{5.11}$$

where $w(Z)$ is a weight factor computed according to

$$w(\psi) = \frac{f_Y(\psi)}{f_Z(\psi)}, \tag{5.12}$$

i.e., the weight factor is the quota of the density function for the population \mathcal{Y} , $f_Y(\psi)$, and the density function of the population \mathcal{Z} , $f_Z(\psi)$. The latter density function is referred to as the *importance sampling function*. This density function may be chosen freely as long as it fulfils the condition that all values that appear in the population \mathcal{Y} should also appear in the population \mathcal{Z} :

$$f_Z(\psi) > 0 \quad \forall \quad \psi: f_Y(\psi) > 0. \tag{5.13}$$

The estimate from simple sampling of $X(Z)$ is also an estimate of $E[X]$, because

$$E[M_{X(Z)}] = \{\text{cf. theorem 4.1}\} = E[X(Z)] = \int_{\psi} f_Z(\psi) w(\psi) g(\psi) d\psi = \int_{\psi} f_Z(\psi) \frac{f_Y(\psi)}{f_Z(\psi)} g(\psi) d\psi =$$

$$= \int_{\psi} f_Y(\psi) g(\psi) d\psi = E[g(Y)]. \quad (5.14)$$

Let us now examine the precision of sampling $X(Z)$:

$$\begin{aligned} \text{Var}[M_{X(Z)}] &= \{\text{cf. theorem 4.2}\} = \frac{\text{Var}[X(Z)]}{n} = \frac{1}{n} (E[(X(Z)^2)] - (E[X(Z)])^2) = \\ &= \frac{1}{n} \left(\int_{\psi} f_Z(\psi) (w(\psi) g(\psi))^2 d\psi - \mu_X^2 \right) = \frac{1}{n} \left(\int_{\psi} f_Z(\psi) \frac{f_Y^2(\psi)}{f_Z^2(\psi)} g^2(\psi) d\psi - \mu_X^2 \right) = \\ &= \frac{1}{n} \left(\int_{\psi} f_Y(\psi) w(\psi) g^2(\psi) d\psi - \mu_X^2 \right). \end{aligned} \quad (5.15)$$

We can see that the precision is depending on the chosen importance sampling function, $f_Z(\psi)$. If we would use the actual probability distribution of the inputs to the system, $f_Y(\psi)$, as importance sampling function then (5.15) yields

$$\begin{aligned} \text{Var}[M_{X(Z)}] &= \frac{1}{n} \left(\int_{\psi} f_Y(\psi) w(\psi) g^2(\psi) d\psi - \mu_X^2 \right) = \{w(\psi) = 1 \forall \psi\} = \frac{1}{n} \left(\int_{\psi} f_Y(\psi) g^2(\psi) d\psi - \mu_X^2 \right) = \\ &= \frac{\text{Var}[X]}{n}, \end{aligned} \quad (5.16)$$

i.e., the same result as for simple sampling, which is not very surprising considering that generating scenarios according to the probability distribution of the inputs to the system is exactly what we do in simple sampling. However, if we instead would choose the importance sampling function

$$f_Z(\psi) = \frac{f_Y(\psi) g(\psi)}{\mu_X} \quad (5.17)$$

then we would get the precision

$$\begin{aligned} \text{Var}[M_{X(Z)}] &= \frac{1}{n} \left(\int_{\psi} f_Y(\psi) w(\psi) g^2(\psi) d\psi - \mu_X^2 \right) = \frac{1}{n} \left(\int_{\psi} f_Y(\psi) \frac{f_Y(\psi)}{f_Z(\psi)} g^2(\psi) (d\psi) d\psi - \mu_X^2 \right) = \\ &= \frac{1}{n} \left(\int_{\psi} f_Y(\psi) \frac{f_Y(\psi) \mu_X}{f_Y(\psi) g(\psi)} g^2(\psi) (d\psi) d\psi - \mu_X^2 \right) = \frac{1}{n} \left(\mu_X \int_{\psi} f_Y(\psi) g(\psi) (d\psi) d\psi - \mu_X^2 \right) = \\ &= \frac{1}{n} (\mu_X \cdot \mu_X - \mu_X^2) = 0. \end{aligned} \quad (5.18)$$

An importance sampling function chosen according to (5.17) is referred to as an optimal importance sampling function, because the result of the simulation will always be exactly equal to the true value of $E[X]$.

A Monte Carlo simulation that always produces the exact answer seems too good to be true, and that is of course also the case—the catch is that in order to find the optimal function, we must know $E[X] = \mu_X$.¹ However, the point of importance sampling is that if we can find an importance sampling function which is sufficiently close to the optimal function, we can get an estimate which has a better precision than the estimate from simple sampling, i.e., $\text{Var}[M_{X(Z)}] < \text{Var}[M_X]$. It should though be noticed that there is also a potential risk in using importance sampling, as a poor choice of importance sampling function can create the opposite result, i.e., $\text{Var}[M_{X(Z)}] > \text{Var}[M_X]$.

Example 5.4 (estimates from importance sampling). To be added: Examples illustrating good and poor importance sampling functions for a small system with a finite set of scenarios...

5.5.2 Application

To be added: Introduction...

1. And if we already know the answer it is not surprising that we can produce the correct answer again...

Multiple Inputs

To be added: Discussion on how to manage systems with multiple inputs...

To be added: Example showing the computation of the weight factor for a simulation with multiple inputs...

Multiple Outputs

To be added: Discussion on how to manage systems with multiple outputs...

Finding an Importance Sampling Function

To be added: Discussion on how to choose the importance sampling function using a simplified model...

Systematical Errors

To be added: Discussion on how inappropriate importance sampling functions can introduce a systematic error, and how this sometimes can be acceptable...

Simulation Procedure

To be added: Overview, equations, block diagram and example...

5.6 Stratified Sampling

The idea of stratified sampling is similar to importance sampling; more samples should be collected from the most interesting scenarios, and consequently observations from different parts of the population need to be assigned different weight factors. However, this idea is implemented differently in stratified sampling compared to importance sampling. Here, the population will be divided into different parts, which are referred to as *strata*, and each stratum will be investigated separately from the others. Results from different strata will then be given different weight in the computation of the final result.

5.6.1 Principle

Assume that we have a population \mathcal{X} , for which we want to compute the expectation value $E[X]$. This population is divided into strata so that each unit in \mathcal{X} belongs to one stratum h . Consequently, if \mathcal{X}_h is a strata (i.e., the subset of units from the population \mathcal{X} which are belonging to stratum h) then the intersection of two strata must be an empty set, i.e.,

$$\mathcal{X}_h \cap \mathcal{X}_k = \emptyset, h \neq k. \quad (5.19)$$

Moreover, the union of all strata must be equal to the population \mathcal{X} :

$$\bigcup_h \mathcal{X}_h = \mathcal{X}. \quad (5.20)$$

For each stratum we can compute a *stratum weight*, ω_h , which is equal to the relative number of units in the stratum or—equivalently—the probability that a randomly chosen unit from the population belongs to this particular stratum:

$$\omega_h = P(X \in \mathcal{X}_h) = \frac{N_h}{N}, \quad (5.21)$$

where N_h is the number of units in stratum h .

Assume that we have independent estimates, m_{Xh} , of the expectation value of each stratum, i.e.,

$$\mu_{Xh} = E[X_h] = \frac{1}{N_h} \sum_{i=1}^{N_h} x_{h,i}, \quad (5.22)$$

then we can estimate $E[X]$ using the average of the m_{Xh} weighted by the stratum weight,

$$m_X = \sum_{h=1}^L \omega_h m_{Xh}, \quad (5.23)$$

where L is the number of strata. This can be verified by studying

$$\begin{aligned} \mathbb{E} \left[\sum_{h=1}^L \omega_h M_{Xh} \right] &= \sum_{h=1}^L \omega_h \mathbb{E}[M_{Xh}] = \{\text{assume that } M_{Xh} \text{ is an unbiased estimate of } \mathbb{E}[X_h]\} = \\ &= \sum_{h=1}^L \omega_h \mathbb{E}[X_h] = \sum_{h=1}^L \frac{N_h}{N} \cdot \frac{1}{N_h} \sum_{i=1}^{N_h} x_{h,i} = \frac{1}{N} \sum_{h=1}^L \sum_{i=1}^{N_h} x_{h,i} = \{\text{summarising all units in all strata is} \\ &\text{the same as summarising all units in the population}\} = \frac{1}{N} \sum_{i=1}^N x_i = \mathbb{E}[X]. \end{aligned} \quad (5.24)$$

Estimates of the expectation value in a stratum can be obtained by simple sampling:

$$m_{Xh} = \frac{1}{n_h} \sum_{i=1}^{n_h} x_{h,i}. \quad (5.25)$$

In some cases, we might even be able to compute the true expectation value according to (5.22) for some strata, and then we will of course use $m_{Xh} = \mu_{Xh}$.

As usual, we are interested in the precision of the estimate and we should therefore study

$$\begin{aligned} \text{Var} \left[\sum_{h=1}^L \omega_h M_{Xh} \right] &= \omega_1^2 \text{Var}[M_{X1}] + \dots + \omega_L^2 \text{Var}[M_{XL}] \\ &+ 2\omega_1\omega_2 \text{Cov}[M_{X1}, M_{X2}] + \dots + 2\omega_{L-1}\omega_L \text{Cov}[M_{XL-1}, M_{XL}] = \{\text{the estimates for each stratum} \\ &\text{are calculated separately from each others and are thus independent}\} = \\ &= \sum_{h=1}^L \omega_h^2 \text{Var}[M_{Xh}]. \end{aligned} \quad (5.26)$$

We can see that the variance of the estimate is depending on the stratification (i.e., the choice of strata). If we would use only one stratum, we have $L = 1$ and $\omega_1 = 1$; the result of (5.26) is then

$$\text{Var} \left[\sum_{h=1}^L \omega_h M_{Xh} \right] = 1 \cdot \text{Var}[M_{Xh}], \quad (5.27)$$

i.e., the same result as for simple sampling, which is not very surprising considering that randomly selecting units from the entire population is exactly what we do in simple sampling. However, if we instead would define strata which are completely homogeneous, i.e.,

$$x_{h,i} = x_{h,j} \quad \forall h, i, j, \quad (5.28)$$

then we would get the precision

$$\text{Var} \left[\sum_{h=1}^L \omega_h M_{Xh} \right] = \sum_{h=1}^L \omega_h^2 \cdot 0 = 0. \quad (5.29)$$

An importance sampling function chosen according to (5.17) is referred to as a perfect stratification, because the result of the simulation will always be exactly equal to the true value of $\mathbb{E}[X]$.

A Monte Carlo simulation that always produces the exact answer seems too good to be true, and that is of course also the case—the catch is that in the perfect stratification, the stratum weights must be equal to the probability that X is equal to a specific value x , i.e., the frequency function of X , and if $f_X(x)$ is known we can compute the expectation value directly instead. However, the point of stratified sampling is that if we can find a stratification where strata are sufficiently homogeneous, we can get an estimate

which has a better precision than the estimate from simple sampling. It should though be noticed that there is also a potential risk in using stratified sampling, as a poor stratification can create the opposite result.

Example 5.5 (estimates from stratified sampling). To be added: Examples illustrating good and poor importance sampling functions for a small system with a finite set of scenarios...

Old example: Consider a six-sided dice, where two sides are labelled “0” and the remaining four sides are labelled “1”. Compare the expectation value and variance of the entire population compared to the properties of two strata (one for the two units with the value 0, and another stratum for the remaining units).

Solution: The population in this example can be described as $\mathcal{X} = \{0, 0, 1, 1, 1, 1\}$, i.e., one unit per side of the dice. However, in order to reduce the length of the computations, we can remove duplicates from the population, i.e., let us consider $\mathcal{X} = \{0, 1, 1\}$. The two strata are then defined by the subsets $\mathcal{X}_1 = \{0\}$ and $\mathcal{X}_2 = \{1, 1\}$. The expectation values and variances can be computed directly according to the definitions:

$$E[X] = (0 + 1 + 1)/3 = 2/3,$$

$$\text{Var}[X] = ((0 - 2/3)^2 + (1 - 2/3)^2 + (1 - 2/3)^2)/3 = 2/9,$$

$$E[X_1] = 0,$$

$$\text{Var}[X_1] = 0,$$

$$E[X_2] = (1 + 1)/2 = 1,$$

$$\text{Var}[X_2] = 0.$$

5.6.2 Application

To be added: Introduction...

Sample Allocation

To be added: Discussion on how to distribute samples between strata...

Theorem 5.2. (Neyman allocation) For a given stratification, the variance of the estimate from stratified sampling, i.e.,

$$\text{Var} \left[\sum_{h=1}^L \omega_h M_{Xh} \right],$$

is minimised if samples are distributed according to

$$n_h = n \frac{\omega_h \sigma_{Xh}}{\sum_{k=1}^L \omega_k \sigma_{Xk}}.$$

Algorithm 5.3. A reasonable compromise allocation for a batch of samples in stratified sampling can be computed as follows:

Step 1. Calculate a compromise allocation, n_h^{\circledast} for $n + n_b$ samples (where n is the total number of samples collected so far and n_b is the number of samples to be collected in batch b).

Step 2. Calculate a preliminary batch allocation according to $n'_{h,b} = n_h^{\circledast} - n_h$, where n_h is the number of samples collected so far from stratum h .

Step 3. Let \mathcal{H}^+ be the index set of strata which should be allocated more samples, i.e.,

$$\mathcal{H}^+ = \{h: n'_{h,b} > 0\}.$$

Step 4. Calculate the total number of samples needed to collect at least as many samples as in the target allocation for each stratum:

$$n^+ = \sum_{h \in \mathcal{H}^+} n'_{h,b}.$$

Step 5. Let \mathcal{H}^- be the index set of strata which have received to many samples, i.e.,

$$\mathcal{H}^- = \{h: n'_{h,b} < 0\}.$$

Step 6. Calculate the total number of additional samples compared to the target allocation for each stratum:

$$n^- = - \sum_{h \in \mathcal{H}^-} n'_{h,b}.$$

Step 7. Calculate the batch allocation according to

$$n_{h,b} = \begin{cases} 0 & \forall h \in \mathcal{H}^-, \\ (1 - n^-/n^+)n'_{h,b} & \forall h \in \mathcal{H}^+. \end{cases}$$

The Cardinal Error

To be added: Discussion on how the practical application of the Neyman allocation may introduce a systematical error when sampling duogeneous populations...

The Cum \sqrt{f} Rule

To be added: Discussion on how to design strata using a simplified model...

The Strata Tree

To be added: Discussion on how to design strata using classification of input scenarios...

Simulation Procedure

To be added: Overview, equations (including random number generation), block diagram and example...

Example 5.6 (random number generation in stratified sampling). To be added...

Exercises

- 5.1 Y is $U(100\ 200)$ -distributed. If $y_i = 125$, then what is the value of the complementary random number y_i^* ?
- 5.2 Figure 5.1 shows the distribution function of the continuous random variable Y . If $y_i = 32$, then what is the value of the complementary random number y_i^* ?
- 5.3 Figure 5.2 shows the distribution function of the continuous random variable Y . Use the following random numbers from a $U(0, 1)$ -distribution to compute a value of Y ; moreover, compute the corresponding complementary random number Y^* .
 - a) 0.80.
 - b) 0.14.
 - c) 0.42.

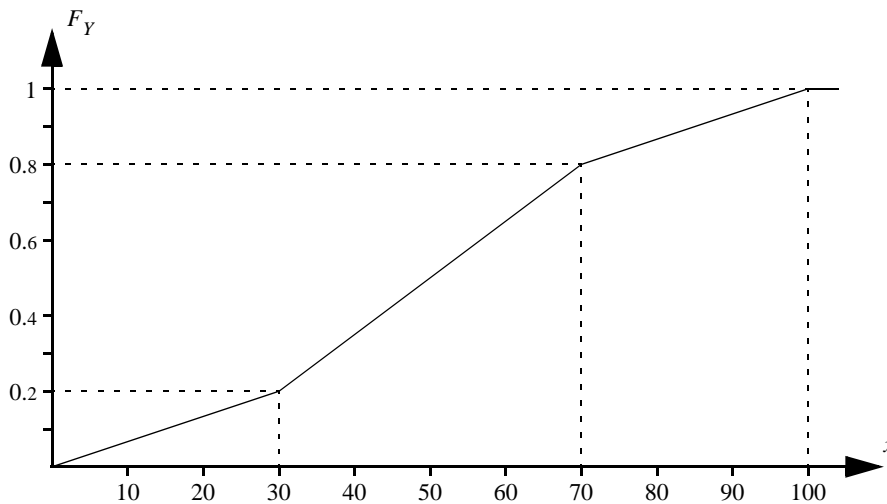


Figure 5.1 Distribution function in exercise 5.2.

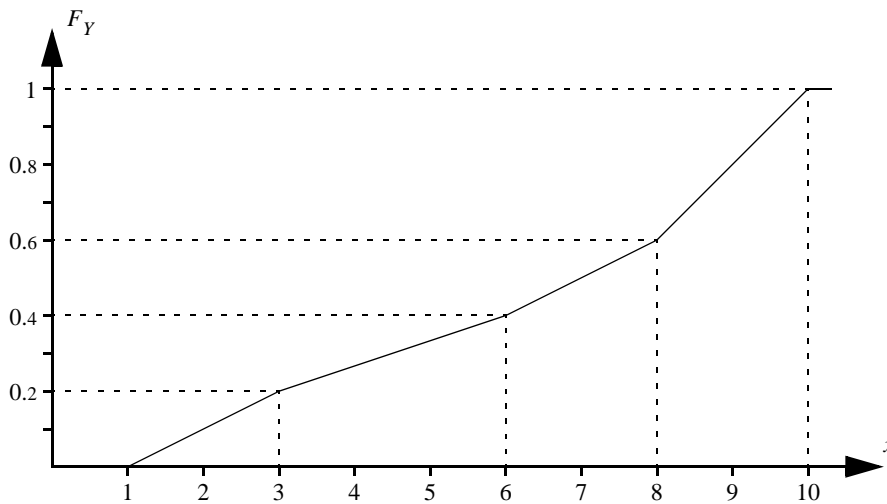


Figure 5.2 Distribution function in exercise 5.3.

5.4 The random variable Y has the frequency function

$$f_Y(x) = \begin{cases} 0.72 & x = 10, \\ 0.28 & x = 20, \\ 0 & \text{all other } x. \end{cases}$$

Apply dagger sampling to generate a sequence of values of Y based on the following random numbers from a $U(0, 1)$ -distribution:

- a) 0.66.
- b) 0.85.
- c) 0.04.

5.5 A system has two binary inputs, Y_1 and Y_2 , such that $P(Y_1 = 0) = 0.87$, $P(Y_1 = 1) = 0.13$, $P(Y_2 = 0) = 0.94$ and $P(Y_2 = 1) = 0.06$. The probability distributions of these two inputs have been combined into a

single joint distribution as shown in figure 5.3. Apply the inverse transform method to convert the random value 0.46 from a $U(0, 1)$ -distribution into values of Y_1 and Y_2 .

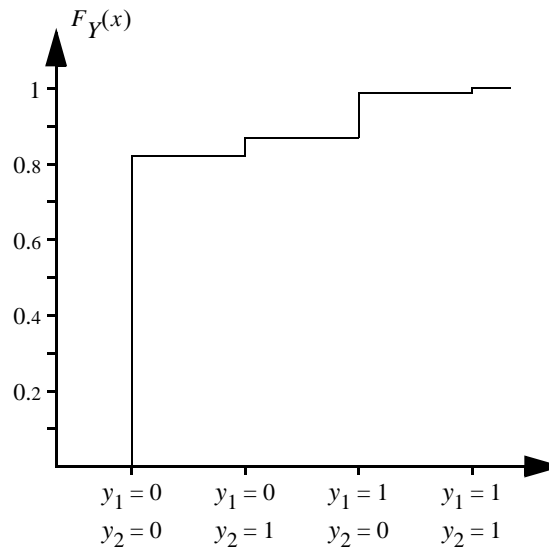


Figure 5.3 Distribution function in exercise 5.5.

- a) Assume that 20 scenarios are to be generated (i.e., we must generate 20 values of Y_1 and 20 values of Y_2). How many values from the random number generator are needed if the inverse transform method (i.e., simple sampling) is applied?
- b) Assume that 20 scenarios are to be generated. How many values from the random number generator are needed if complementary random numbers are applied?
- c) Assume that 20 scenarios are to be generated. How many values from the random number generator if dagger sampling is applied? Assume that the dagger cycles for the two inputs are independent.
- d) Assume that 20 scenarios are to be generated. How many values from the random number generator if dagger sampling is applied? Assume that the dagger cycles for the two inputs are reset at the end of the shorter cycle.
- e) Assume that 20 scenarios are to be generated. How many values from the random number generator if dagger sampling is applied? Assume that the dagger cycles for the two inputs are reset at the end of the longer cycle.

5.6 The random variable Y has the frequency function

$$f_Y(x) = \begin{cases} 0.15 & x = 0, \\ 0.85 & x = 1, \\ 0 & \text{all other } x. \end{cases}$$

- a) Assume that six random values of Y are generated. What is the probability that all six values are equal to one (i.e., $y_i = 1, i = 1, \dots, 6$) if the values are generated independently using the inverse transform method (i.e., simple sampling)?
- b) Assume that six random values of Y are generated. What is the probability that all six values are equal to one if the values are generated independently using complementary random numbers?
- c) Assume that six random values of Y are generated. What is the probability that all six values are equal to one if the values are generated independently using dagger sampling?

5.7 The random variable Y has the frequency function

$$f_Y(x) = \begin{cases} 0.1 & x = 0, \\ 0.9 & x = 1, \\ 0 & \text{all other } x. \end{cases}$$

Table 5.4 Random numbers from a $U(0, 1)$ distribution

| | | | | |
|------|------|------|------|------|
| 0.44 | 0.77 | 0.19 | 0.45 | 0.71 |
| 0.38 | 0.80 | 0.49 | 0.65 | 0.75 |

- a) Apply the inverse transform method to generate 10 values of Y . This problem should be solved using all the random numbers in table 5.4.
- b) Apply complementary random numbers to generate 10 values of Y . This problem should be solved using the first row of the random numbers in table 5.4.
- c) Apply dagger sampling to generate 10 values of Y . This problem should be solved using as many of the random numbers in table 5.4 as needed.
- d) Compare the results of the previous questions. Which sequence of random numbers is closest to the actual probability distribution? Is this a coincidence or is there an explanation to the result?

5.8 Consider a system with three inputs,

$$X = g(Y_1, Y_2, Y_3) = \begin{cases} 0 & \text{if } Y_1(Y_2 + Y_3) > 0, \\ 1 & \text{if } Y_1(Y_2 + Y_3) = 0. \end{cases}$$

The frequency function of each input is

$$f_Y(x) = \begin{cases} 0.1 & x = 0, \\ 0.9 & x = 1, \\ 0 & \text{all other } x. \end{cases}$$

The system is simulated using the importance sampling function

$$f_Z(x) = \begin{cases} 0.4 & x = 0, \\ 0.6 & x = 1, \\ 0 & \text{all other } x, \end{cases}$$

for each component. Table 5.5 shows five scenarios for this system. Calculate the estimated expectation value of X .

Table 5.5 Five scenarios for the Monte Carlo simulation in exercise 5.8.

| Scenario | Y_1 | Y_2 | Y_3 |
|----------|-------|-------|-------|
| 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 1 |
| 3 | 1 | 0 | 1 |
| 4 | 1 | 1 | 1 |
| 5 | 1 | 1 | 0 |

5.9 A system has two inputs and one output. Table 5.6 shows the results from a Monte Carlo simulation of this system using importance sampling. What are the estimates of $E[X]$ and $\text{Var}[X]$ respectively?

Table 5.6 Results from the Monte Carlo simulation in exercise 5.9.

| Scenario, i | Input 1 | | | Input 2 | | | Output value, x_i |
|---------------|------------------|--------------------------------------|--|------------------|--------------------------------------|--|---------------------|
| | Value, $y_{1,i}$ | Density function, $f_{Y_1}(y_{1,i})$ | Importance sampling function, $f_{Z_1}(y_{1,i})$ | Value, $y_{2,i}$ | Density function, $f_{Y_2}(y_{2,i})$ | Importance sampling function, $f_{Z_2}(y_{2,i})$ | |
| 1 | 90 | 1/100 | 1/40 | 99 | 1/100 | 1/20 | 379 |
| 2 | 20 | 1/100 | 1/160 | 91 | 1/100 | 1/20 | 202 |
| 3 | 81 | 1/100 | 1/40 | 12 | 1/100 | 1/180 | 173 |
| 4 | 87 | 1/100 | 1/40 | 13 | 1/100 | 1/180 | 197 |
| 5 | 96 | 1/100 | 1/40 | 23 | 1/100 | 1/180 | 215 |

5.10 Consider a system $X = g(Y_1, Y_2)$, where

$$g(Y_1, Y_2) = \begin{cases} Y_1 + Y_2 & \text{if } Y_1 + Y_2 < 10, \\ 2(Y_1 + Y_2) & \text{if } 10 \leq Y_1 + Y_2 < 20, \\ 3.25(Y_1 + Y_2) & \text{if } 20 \leq Y_1 + Y_2. \end{cases}$$

The inputs are discrete random variables, and each input is either equal to 0, 5 or 10. The probability of each state is recorded in table 5.7.

Table 5.7 Probability distribution of the inputs in exercise 5.10.

| State, x | $f_{Y_1}(x)$ | $f_{Y_2}(x)$ |
|------------|--------------|--------------|
| 0 | 0.25 | 0.20 |
| 5 | 0.50 | 0.60 |
| 10 | 0.25 | 0.20 |

What is the optimal importance sampling function for this system?

5.11 Consider a simplified model

$$Z = \tilde{g}(Y) = \begin{cases} Y & \text{if } Y \leq 10, \\ 2Y - 10 & \text{if } Y > 10, \end{cases}$$

of the system $X = g(Y)$, where the input Y has the density function

$$f_Y(x) = \begin{cases} 0 & x < 0, \\ 0.05 & 0 \leq x \leq 5, \\ 0.10 & 5 < x \leq 10, \\ 0.05 & 10 < x \leq 15, \\ 0 & 15 < x. \end{cases}$$

Use the simplified model to suggest an importance sampling function for this system.

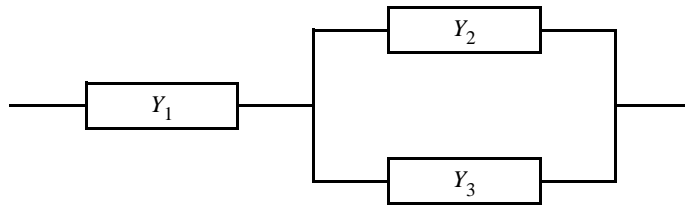


Figure 5.4 The system in exercise 5.13.

5.12 Table 5.8 lists the possible states of a simplified model of a system with two outputs. Suggest an importance sampling function for this system.

Table 5.8 Scenarios for the system in exercise 5.12.

| Input, y | Probability $f_Y(y)$ | Outputs from the simplified model | |
|---------------|-------------------------|-----------------------------------|-------|
| | | z_1 | z_2 |
| 1 | 0.1 | 100 | 0 |
| 2 | 0.2 | 225 | 0 |
| 3 | 0.4 | 350 | 0 |
| 4 | 0.2 | 575 | 0 |
| 5 | 0.1 | 900 | 1 |

5.13 Consider the system in figure 5.4, where each component has a reliability of 90%. The inputs Y_1 , Y_2 and Y_3 are equal to 1 if a component is operational, and 0 otherwise. The output X is equal to 1 if the system is working, and 0 if it is not, i.e.,

$$X = \begin{cases} 1 & \text{if } Y_1 = 1 \text{ and } Y_2 + Y_3 \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

The system is to be simulated using importance sampling. Is it more efficient to use the importance sampling function

$$f_{Z1}(y) = \begin{cases} 0.5 & y = 0, \\ 0.5 & y = 1, \\ 0 & \text{all other } y, \end{cases}$$

for each component, or to use the importance sampling function

$$f_{Z2}(y) = \begin{cases} 0.2 & y = 0, \\ 0.8 & y = 1, \\ 0 & \text{all other } y, \end{cases}$$

for each component? The computation time is the same for both importance sampling functions.

5.14 Table 5.9 shows the results of the first 1 000 samples from a Monte Carlo simulation.

Table 5.9 Results from the Monte Carlo simulation in exercise 5.14.

| Stratum, h | Stratum weight, ω_h | Number of samples, n_h | Simulation results | |
|--------------|----------------------------|--------------------------|----------------------------|------------------------------|
| | | | $\sum_{h=1}^{n_h} x_{h,i}$ | $\sum_{h=1}^{n_h} x_{h,i}^2$ |
| 1 | 0.4 | 250 | 50 000 | 260 000 000 |
| 2 | 0.3 | 400 | 200 000 | 1 000 000 000 |
| 3 | 0.2 | 300 | 165 000 | 1 009 500 000 |
| 4 | 0.1 | 50 | 50 000 | 62 500 000 |

- a) Calculate the estimated expectation value.
- b) Suggest a sample allocation for the next batch of 1 000 samples.

5.15 Table 5.10 shows the results of a pilot study for a Monte Carlo simulation of the model $g(Y_1, Y_2)$. Suggest a sample allocation for the next batch of 100 samples.

Table 5.10 Results from the Monte Carlo simulation in exercise 5.15 after a pilot study of 100 scenarios.

| Stratum, h | Number of samples, n_h | Estimated stratum mean, m_{Xh} | Estimated stratum variance, s_{Xh}^2 |
|--------------|--------------------------|----------------------------------|--|
| 1 | 25 | 60 | 1 024 |
| 2 | 25 | 65 | 529 |
| 3 | 25 | 90 | 1 936 |
| 4 | 25 | 110 | 3 600 |

5.16 Table 5.11 shows the results of the first 1 000 samples from a Monte Carlo simulation of a model with two outputs. Suggest an appropriate sample allocation for the next batch of 1 000 scenarios.

Table 5.11 Results from the Monte Carlo simulation in exercise 5.16.

| Stratum, h | Stratum weight, ω_h | Number of samples, n_h | Estimated stratum mean, m_{Xh} | | Estimated stratum variance, s_{Xh}^2 | |
|--------------|----------------------------|--------------------------|----------------------------------|----------|--|----------|
| | | | Output 1 | Output 2 | Output 1 | Output 2 |
| 1 | 0.5 | 200 | 200 | 0 | 4 900 | 0 |
| 2 | 0.3 | 400 | 260 | 12 | 22 500 | 9.00 |
| 3 | 0.2 | 400 | 290 | 45 | 10 000 | 30.25 |

5.17 Y_1 and Y_2 are two independent random variables with distribution functions $F_{Y_1}(x)$ and $F_{Y_2}(x)$ according to figure 5.5. The system $X = g(Y_1, Y_2)$ is simulated using stratified sampling. Strata have been defined according to figure 5.6.

- a) Calculate the stratum weights.
- b) Table 5.12 shows ten scenarios in a Monte Carlo simulation of the model $g(Y_1, Y_2)$. Calculate the estimated expectation value according to these samples.

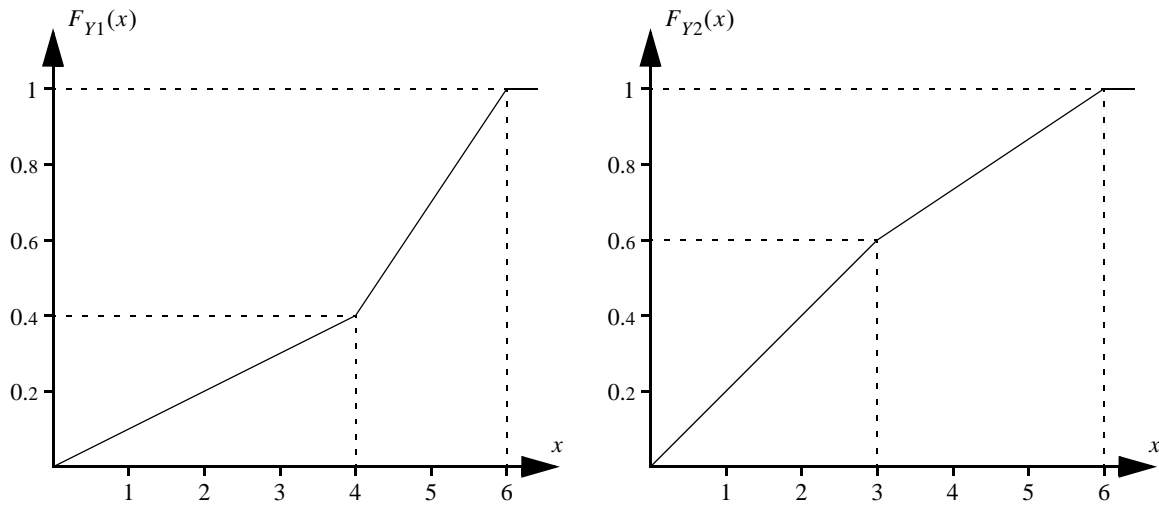


Figure 5.5 Probability distributions of the inputs in exercise 5.17.

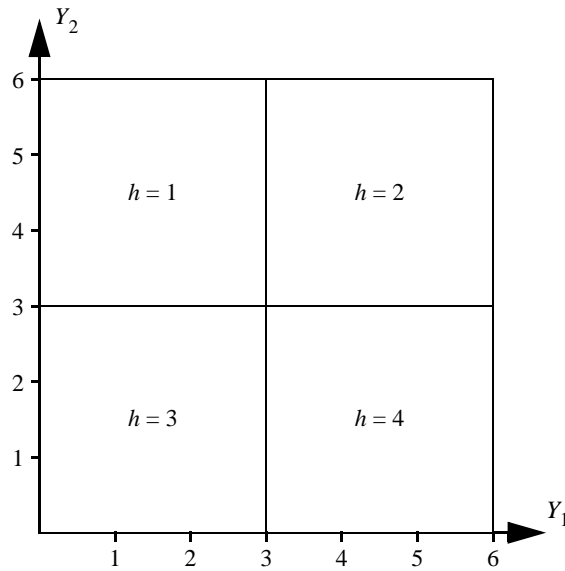
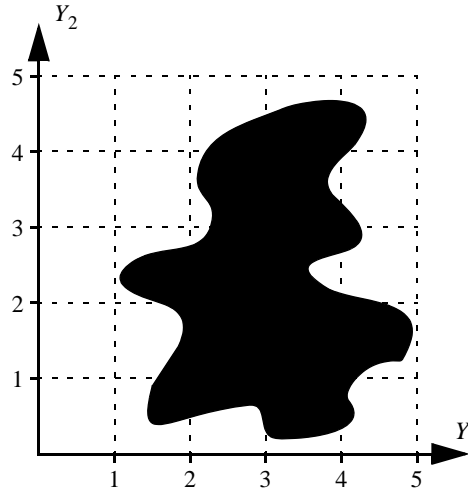


Figure 5.6 Strata in exercise 5.17.

Table 5.12 Results from the Monte Carlo simulation in exercise 5.17.

| Scenario | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------|-----|-----|-----|-----|------|------|-----|------|-----|------|
| Y_1 | 0.5 | 1.2 | 1.6 | 2.8 | 3.4 | 4.4 | 4.5 | 5.2 | 5.5 | 5.9 |
| Y_2 | 1.3 | 4.8 | 2.2 | 5.2 | 5.1 | 3.9 | 2.4 | 4.3 | 0.6 | 3.6 |
| X | 2.1 | 5.7 | 3.3 | 8.5 | 14.7 | 13.6 | 7.3 | 16.9 | 5.7 | 15.6 |

5.18 The area of the shape in the figure below is to be estimated using stratified sampling.



- a) Define appropriate strata.
- b) Calculate the corresponding stratum weights.

5.19 Alice is using her car to travel to work. She can choose between two alternative routes. Route is the shorter one, but traffic might on the other hand be busy. Route B is longer, but there are hardly any unforeseen traffic jams. In order to decide which way to go, Alice listens to the traffic reports on the radio and estimates the travel time for each road, and then she chooses the route with the shortest forecasted travel time. If the forecasted travel time is the same for both routes, she will choose route A. The real travel time might of course be different from the forecast, and it might turn out that the route she has chosen is slower (but then there is no turning back).

The forecasted and real travel times on route A are correlated, but can be considered independent if we separate between low and busy traffic as well as accurate and inaccurate forecast, as described in the table below. The forecasted travel time for route B is always $U(45, 75)$ -distributed and the real travel time is also $U(45, 75)$ -distributed. The forecasted and real travel time on route B can be considered independent.

Table 5.13 Overview of forecasted and real travel times on route A.

| Situation | Probability [%] | Forecasted travel time [min] | Real travel time [min] |
|-----------------------------------|-----------------|------------------------------|------------------------|
| Low traffic, accurate forecast | 60 | $U(30, 40)$ | $U(30, 40)$ |
| Busy traffic, inaccurate forecast | 10 | $U(30, 40)$ | $U(60, 90)$ |
| Low traffic, inaccurate forecast | 2 | $U(60, 90)$ | $U(30, 40)$ |
| Busy traffic, accurate forecast | 28 | $U(60, 90)$ | $U(60, 90)$ |

- a) Suggest appropriate strata for a simulation of Alice’s travelling, where the objective of the simulation is to estimate the expected travel time.
 - b) Assume that the simulation should start with a pilot study and that 100 scenarios generated in the pilot study. How would it be appropriate to distribute the scenarios in the pilot study among the suggested strata?
- 5.20 Consider a system $X = g(Y_1, Y_2)$. The inputs Y_1 and Y_2 are independent and there are six possible states for this system, as listed in table 5.14.
- Assume that this system is simulated using stratified sampling. Two strata have been defined so that all scenarios where $Y_2 = 0$ belong to stratum 1 and all scenarios where $Y_2 = 2$ belongs to stratum 2. Moreover, assume that $n/2$ scenarios will be generated in stratum 1 and $n/2$ scenarios in stratum 2.

- a) Is this application of stratified sampling more efficient than simple sampling with n samples?

b) Can the efficiency of stratified sampling be improved further? If that is the case, what should be done?

Table 5.14 Possible states of the system in exercise 5.20.

| y_1 | y_2 | $f_{Y_1}(x)$ | $f_{Y_2}(x)$ | $g(y_1, y_2)$ |
|-------|-------|--------------|--------------|---------------|
| 1 | 0 | 0.2 | 0.01 | 5 |
| 1 | 2 | 0.2 | 0.99 | 1 |
| 2 | 0 | 0.6 | 0.01 | 10 |
| 2 | 2 | 0.6 | 0.99 | 2 |
| 3 | 0 | 0.2 | 0.01 | 15 |
| 3 | 2 | 0.2 | 0.99 | 7 |

5.21 Table 5.15 shows the results from a Monte Carlo simulation. Complementary random numbers are applied to one of the inputs of the systems, i.e., for each scenario y_i that is generated, we also get a complementary scenario y_i^* . A simplified model is used to generate control variates, and the expectation value of the simplified model has been calculated to 9. Moreover, stratified sampling has been used. 100 original and 100 complementary scenarios have been generated for each stratum. Calculate the estimate of $E[X]$.

Table 5.15 Results from the Monte Carlo simulation in exercise 5.21.

| Stratum, h | Stratum weight, ω_h | Results from detailed model | | Results from simplified model | |
|--------------|----------------------------|--|---|--|---|
| | | Original scenarios, 100 $\sum_{i=1}^{100} x_{h,i}$ | Complementary scenarios, 100 $\sum_{i=1}^{100} x_{h,i}^*$ | Original scenarios, 100 $\sum_{i=1}^{100} z_{h,i}$ | Complementary scenarios, 100 $\sum_{i=1}^{100} z_{h,i}^*$ |
| 1 | 0.5 | 1 600 | 1 800 | 1 500 | 1 650 |
| 2 | 0.3 | 2 400 | 2 200 | 1 900 | 1 800 |
| 3 | 0.2 | 3 000 | 3 200 | 2 200 | 2 350 |

5.22 The objective of a simulation is to compare the two systems $X_1 = g_1(Y)$ and $X_2 = g_2(Y)$. The systems have been simulated using correlated sampling, complementary random numbers (which however is applied only to one of the inputs of the systems) and importance sampling. Introduce the following symbols:

- $f_Z(x)$ = importance sampling function,
- $f_Y(x)$ = density function of the input variables,
- y_i = outcome of the input variables in the i :th scenario,
- w_i = weight factor of scenario y_i , i.e., $f_Y(y_i)/f_Z(y_i)$,
- $x_{1,i}$ = output of system 1 for the i :th scenario, i.e., $g_1(y_i)$,
- $x_{2,i}$ = output of system 2 for the i :th scenario, i.e., $g_2(y_i)$,
- y_i^* = complementary scenario of y_i ,
- w_i^* = weight factor of the complementary scenario y_i^* , i.e., $f_Y(y_i^*)/f_Z(y_i^*)$,
- $x_{1,i}^*$ = output of system 1 for the complementary scenario y_i^* , i.e., $g_1(y_i^*)$,
- $x_{2,i}^*$ = output of system 2 for the complementary scenario y_i^* , i.e., $g_2(y_i^*)$.

The following results have been obtained:

$$\sum_{i=1}^{500} w_i x_{1,i} = 50\,300, \quad i = 1, \dots, 500,$$

$$\sum_{i=1}^{500} w_i^* x_{1,i}^* = 49\,600, \quad i = 1, \dots, 500,$$

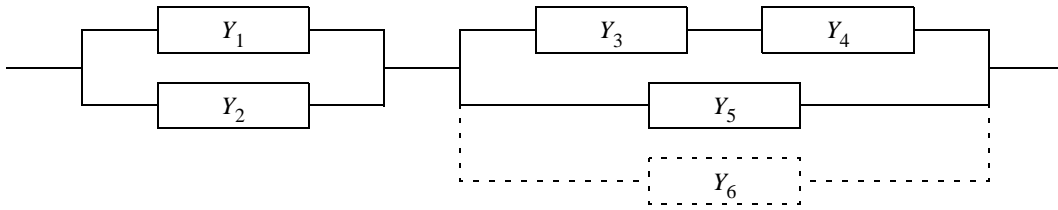


Figure 5.7 The system in exercise 5.24.

$$\sum_{i=1}^{500} w_i x_{2,i} = 49\,100, \quad i = 1, \dots, 500,$$

$$\sum_{i=1}^{500} w_i^* x_{2,i}^* = 48\,800, \quad i = 1, \dots, 500.$$

Calculate the estimated expected difference between the two systems.

- 5.23 Consider a Monte Carlo simulation which is using stratified sampling, complementary random numbers (on one input) and importance sampling (on the remaining inputs). The results of the simulation are shown in table 5.16. What is the estimated expectation value of this system?

Table 5.16 Results from the Monte Carlo simulation in exercise 5.23.

| Stratum, h | Stratum weight, ω_h | Samples per stratum, n_h | Results from original scenarios, $\sum_{i=1}^{n_h/2} \frac{f_Y(y_{h,i})}{f_Z(y_{h,i})} g(y_{h,i})$ | Results from complementary scenarios, $\sum_{i=1}^{n_h/2} \frac{f_Y(y_{h,i}^*)}{f_Z(y_{h,i}^*)} g(y_{h,i}^*)$ |
|--------------|----------------------------|----------------------------|---|--|
| 1 | 0.5 | 200 | 1 650 | 1 550 |
| 2 | 0.3 | 400 | 3 800 | 4 600 |
| 3 | 0.2 | 400 | 7 400 | 6 200 |

- 5.24 Consider the system in figure 5.7. In its original configuration it has five components. Now it is considered to add some extra redundancy to the system, by adding a backup component for components 3–5. Each component has a reliability of 90% and the state of each component is an input to the system. The inputs are equal to 1 if the corresponding component is operational, and 0 otherwise. The output is equal to 1 if the system is working, and 0 if it is not. Thus, for the original configuration we get the model

$$X_1 = g_1(Y) = \begin{cases} 1 & \text{if } (Y_1 + Y_2)(Y_3 Y_4 + Y_5) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, for the second model we get

$$X_2 = g_2(Y) = \begin{cases} 1 & \text{if } (Y_1 + Y_2)(Y_3 Y_4 + Y_5 + Y_6) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

The improved reliability of the system is to be estimated using correlated sampling and importance sampling. The importance sampling function is set up so that the probability of components 3–5 being operational is set to 50% each, whereas the other components still each have 90% probability of working, i.e.,

$$f_{Z1}(x) = f_{Y1}(x), f_{Z2}(x) = f_{Y2}(x), f_{Z3}(x) = f_{Z4}(x) = f_{Z5}(x) = \begin{cases} 0.5 & x = 0, \\ 0.5 & x = 1, \quad \text{and } f_{Z6}(x) = f_{Y6}(x). \\ 0 & \text{otherwise,} \end{cases}$$

Table 5.17 shows ten scenarios generated using these importance sampling functions. Based on these scenarios, what is the estimated increase in reliability by adding the sixth component?

Table 5.17 Ten scenarios from the Monte Carlo simulation in exercise 5.24.

| Scenario | Component 1 | Component 2 | Component 3 | Component 4 | Component 5 | Component 6 |
|----------|-------------|-------------|-------------|-------------|-------------|-------------|
| 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 2 | 1 | 1 | 0 | 1 | 1 | 1 |
| 3 | 1 | 0 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 0 | 1 | 0 |
| 5 | 1 | 0 | 0 | 1 | 1 | 1 |
| 6 | 1 | 0 | 0 | 0 | 1 | 1 |
| 7 | 1 | 1 | 0 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 0 | 0 | 1 |
| 9 | 1 | 1 | 1 | 0 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 |

EFFICIENT MONTE CARLO SIMULATIONS

The previous chapters have presented the mathematics of Monte Carlo simulation as well as some practical solutions to implement different methods when simulating a technical system. In this concluding chapter, all those pieces are brought together in a discussion on how to design an efficient Monte Carlo simulation.

6.1 Mathematical Model

To be added: Discussion on important steps when formulating the mathematical model...

6.2 Choice of Simulation Method

To be added: Summary of the information necessary to efficiently apply different variance reduction technique.

To be added: Discussion on how variance reduction techniques can be combined.

6.3 Testing

To be added: Discussion on how to test and verify the results of a Monte Carlo simulation...

PROBABILITY DISTRIBUTIONS

This appendix provides an overview of some important probability distributions.
To be added: Reference to other sources.

AKABUGA DISTRICT SIMULATION

The simulation methods presented in this compendium are demonstrated in several examples describing a simulation of the power system in the fictitious East African area Akabuga District. This appendix provides further details of the Akabuga District simulation, which may help the reader understand the results of the different simulation strategies.

B.1 Analysis of the System

To be added: Overview...

System Data

The values of constant parameters in the model of Akabuga District are listed in table B.1. The probability distributions of the inputs are listed below:

$f_{\bar{G}_1}$ = available generation capacity in the large diesel generator set =

$$= \begin{cases} 0.1 & x = 0, \\ 0.9 & x = 200, \\ 0 & \text{all other } x, \end{cases}$$

$f_{\bar{G}_2}$ = available generation capacity in the large diesel generator set =

$$= \begin{cases} 0.2 & x = 0, \\ 0.8 & x = 150, \\ 0 & \text{all other } x, \end{cases}$$

$f_{\bar{P}}$ = available generation capacity in the large diesel generator set =

$$= \begin{cases} 0.01 & x = 0, \\ 0.99 & x = 300, \\ 0 & \text{all other } x, \end{cases}$$

$$f_{D_{tot}} = \text{total demand} = \begin{cases} 0.2 & x = 200, \\ 0.4 & x = 300, \\ 0.25 & x = 400, \\ 0.1 & x = 500, \\ 0.05 & x = 600, \\ 0 & \text{all other } x, \end{cases}$$

$$f_C = \text{the share of the total demand located in Akabuga} = \begin{cases} 0.5 & x = 0.85, \\ 0.5 & x = 0.9, \\ 0 & \text{all other } x. \end{cases}$$

Table B.1 Model constants in the Akabuga District simulation.

| Symbol | Explanation | Value |
|--------------|---|--------------------------|
| β_{G1} | Variable operation cost of the large diesel generator set | 10 ₪/kWh |
| β_{G2} | Variable operation cost of the small diesel generator set | 12 ₪/kWh |
| γ_L | Loss coefficient for the transmission line | 0.00005 kW ⁻¹ |
| \bar{H} | Available generation capacity in the hydro power plant | 350 kWh/h |

Table B.2 Outputs in the Akabuga District simulation.

| Symbol | Explanation |
|--------|--|
| D_1 | Load in Akabuga [kWh/h] |
| D_2 | Load in Ekyaalo [kWh/h] |
| G_1 | Generation in the large diesel generator set [kWh/h] |
| G_2 | Generation in the small diesel generator set [kWh/h] |
| H | Generation in the hydro power plant [kWh/h] |
| $LOLO$ | Loss of load occasion (1 if load shedding occurs, 0 otherwise) |
| P | Transmission from Ekyaalo to Akabuga [kWh/h] |
| TOC | Total operation cost of the system [₪/h] |
| U | Unserved load [kWh/h] |

Enumeration of Scenarios

Table B.3 lists all possible scenarios for Akabuga District.

To be added: Comments to the table...

Table B.3 Enumeration of scenarios in the Akabuga District simulation.

| Scenario | \bar{P} | \bar{G}_1 | \bar{G}_2 | D_{tot} | D_1 | D_2 | TOC | $LOLO$ | f_Y | $f_Y TOC$ | $f_Y LOLO$ |
|----------|-----------|-------------|-------------|-----------|-------|-------|----------|--------|-----------|-----------|------------|
| 1 | 0 | 0 | 0 | 200 | 170 | 30 | 0 | 1 | 0.000 020 | 0 | 0.000 020 |
| 2 | 0 | 0 | 0 | 200 | 180 | 20 | 0 | 1 | 0.000 020 | 0 | 0.000 020 |
| 3 | 0 | 0 | 0 | 300 | 255 | 45 | 0 | 1 | 0.000 040 | 0 | 0.000 040 |
| 4 | 0 | 0 | 0 | 300 | 270 | 30 | 0 | 1 | 0.000 040 | 0 | 0.000 040 |
| 5 | 0 | 0 | 0 | 400 | 340 | 60 | 0 | 1 | 0.000 025 | 0 | 0.000 025 |
| 6 | 0 | 0 | 0 | 400 | 360 | 40 | 0 | 1 | 0.000 025 | 0 | 0.000 025 |
| 7 | 0 | 0 | 0 | 500 | 425 | 75 | 0 | 1 | 0.000 010 | 0 | 0.000 010 |
| 8 | 0 | 0 | 0 | 500 | 450 | 50 | 0 | 1 | 0.000 010 | 0 | 0.000 010 |
| 9 | 0 | 0 | 0 | 600 | 510 | 90 | 0 | 1 | 0.000 005 | 0 | 0.000 005 |
| 10 | 0 | 0 | 0 | 600 | 540 | 60 | 0 | 1 | 0.000 005 | 0 | 0.000 005 |
| 11 | 0 | 0 | 150 | 200 | 170 | 30 | 1 800.00 | 1 | 0.000 080 | 0.1440 | 0.000 080 |
| 12 | 0 | 0 | 150 | 200 | 180 | 20 | 1 800.00 | 1 | 0.000 080 | 0.1440 | 0.000 080 |
| 13 | 0 | 0 | 150 | 300 | 255 | 45 | 1 800.00 | 1 | 0.000 160 | 0.2880 | 0.000 160 |
| 14 | 0 | 0 | 150 | 300 | 270 | 30 | 1 800.00 | 1 | 0.000 160 | 0.2880 | 0.000 160 |
| 15 | 0 | 0 | 150 | 400 | 340 | 60 | 1 800.00 | 1 | 0.000 100 | 0.1800 | 0.000 100 |
| 16 | 0 | 0 | 150 | 400 | 360 | 40 | 1 800.00 | 1 | 0.000 100 | 0.1800 | 0.000 100 |
| 17 | 0 | 0 | 150 | 500 | 425 | 75 | 1 800.00 | 1 | 0.000 040 | 0.0720 | 0.000 040 |
| 18 | 0 | 0 | 150 | 500 | 450 | 50 | 1 800.00 | 1 | 0.000 040 | 0.0720 | 0.000 040 |
| 19 | 0 | 0 | 150 | 600 | 510 | 90 | 1 800.00 | 1 | 0.000 020 | 0.0360 | 0.000 020 |
| 20 | 0 | 0 | 150 | 600 | 540 | 60 | 1 800.00 | 1 | 0.000 020 | 0.0360 | 0.000 020 |

Table B.3 Enumeration of scenarios in the Akabuga District simulation.

| Scenario | \bar{P} | \bar{G}_1 | \bar{G}_2 | D_{tot} | D_1 | D_2 | TOC | LOLO | f_Y | $f_Y TOC$ | $f_Y LOLO$ |
|----------|-----------|-------------|-------------|-----------|-------|-------|----------|------|-----------|-----------|------------|
| 21 | 0 | 200 | 0 | 200 | 170 | 30 | 1 700.00 | 0 | 0.000 180 | 0.3060 | 0 |
| 22 | 0 | 200 | 0 | 200 | 180 | 20 | 1 800.00 | 0 | 0.000 180 | 0.3240 | 0 |
| 23 | 0 | 200 | 0 | 300 | 255 | 45 | 2 000.00 | 1 | 0.000 360 | 0.7200 | 0.000 360 |
| 24 | 0 | 200 | 0 | 300 | 270 | 30 | 2 000.00 | 1 | 0.000 360 | 0.7200 | 0.000 360 |
| 25 | 0 | 200 | 0 | 400 | 340 | 60 | 2 000.00 | 1 | 0.000 225 | 0.4500 | 0.000 225 |
| 26 | 0 | 200 | 0 | 400 | 360 | 40 | 2 000.00 | 1 | 0.000 225 | 0.4500 | 0.000 225 |
| 27 | 0 | 200 | 0 | 500 | 425 | 75 | 2 000.00 | 1 | 0.000 090 | 0.1800 | 0.000 090 |
| 28 | 0 | 200 | 0 | 500 | 450 | 50 | 2 000.00 | 1 | 0.000 090 | 0.1800 | 0.000 090 |
| 29 | 0 | 200 | 0 | 600 | 510 | 90 | 2 000.00 | 1 | 0.000 045 | 0.0900 | 0.000 045 |
| 30 | 0 | 200 | 0 | 600 | 540 | 60 | 2 000.00 | 1 | 0.000 045 | 0.0900 | 0.000 045 |
| 31 | 0 | 200 | 150 | 200 | 170 | 30 | 1 700.00 | 0 | 0.000 720 | 1.2240 | 0 |
| 32 | 0 | 200 | 150 | 200 | 180 | 20 | 1 800.00 | 0 | 0.000 720 | 1.2960 | 0 |
| 33 | 0 | 200 | 150 | 300 | 255 | 45 | 2 660.00 | 0 | 0.001 440 | 3.8304 | 0 |
| 34 | 0 | 200 | 150 | 300 | 270 | 30 | 2 840.00 | 0 | 0.001 440 | 4.0896 | 0 |
| 35 | 0 | 200 | 150 | 400 | 340 | 60 | 3 680.00 | 0 | 0.000 900 | 3.3120 | 0 |
| 36 | 0 | 200 | 150 | 400 | 360 | 40 | 3 800.00 | 1 | 0.000 900 | 3.4200 | 0.000 900 |
| 37 | 0 | 200 | 150 | 500 | 425 | 75 | 3 800.00 | 1 | 0.000 360 | 1.3680 | 0.000 360 |
| 38 | 0 | 200 | 150 | 500 | 450 | 50 | 3 800.00 | 1 | 0.000 360 | 1.3680 | 0.000 360 |
| 39 | 0 | 200 | 150 | 600 | 510 | 90 | 3 800.00 | 1 | 0.000 180 | 0.6840 | 0.000 180 |
| 40 | 0 | 200 | 150 | 600 | 540 | 60 | 3 800.00 | 1 | 0.000 180 | 0.6840 | 0.000 180 |
| 41 | 300 | 0 | 0 | 200 | 170 | 30 | 0 | 0 | 0.001 980 | 0 | 0 |
| 42 | 300 | 0 | 0 | 200 | 180 | 20 | 0 | 0 | 0.001 980 | 0 | 0 |
| 43 | 300 | 0 | 0 | 300 | 255 | 45 | 0 | 0 | 0.003 960 | 0 | 0 |
| 44 | 300 | 0 | 0 | 300 | 270 | 30 | 0 | 0 | 0.003 960 | 0 | 0 |
| 45 | 300 | 0 | 0 | 400 | 340 | 60 | 0 | 1 | 0.002 475 | 0 | 0.002 475 |
| 46 | 300 | 0 | 0 | 400 | 360 | 40 | 0 | 1 | 0.002 475 | 0 | 0.002 475 |
| 47 | 300 | 0 | 0 | 500 | 425 | 75 | 0 | 1 | 0.000 990 | 0 | 0.000 990 |
| 48 | 300 | 0 | 0 | 500 | 450 | 50 | 0 | 1 | 0.000 990 | 0 | 0.000 990 |
| 49 | 300 | 0 | 0 | 600 | 510 | 90 | 0 | 1 | 0.000 495 | 0 | 0.000 495 |
| 50 | 300 | 0 | 0 | 600 | 540 | 60 | 0 | 1 | 0.000 495 | 0 | 0.000 495 |
| 51 | 300 | 0 | 150 | 200 | 170 | 30 | 0 | 0 | 0.007 920 | 0 | 0 |
| 52 | 300 | 0 | 150 | 200 | 180 | 20 | 0 | 0 | 0.007 920 | 0 | 0 |
| 53 | 300 | 0 | 150 | 300 | 255 | 45 | 0 | 0 | 0.015 840 | 0 | 0 |
| 54 | 300 | 0 | 150 | 300 | 270 | 30 | 0 | 0 | 0.015 840 | 0 | 0 |
| 55 | 300 | 0 | 150 | 400 | 340 | 60 | 650.46 | 0 | 0.009 900 | 6.4396 | 0 |
| 56 | 300 | 0 | 150 | 400 | 360 | 40 | 774.00 | 0 | 0.009 900 | 7.6626 | 0 |
| 57 | 300 | 0 | 150 | 500 | 425 | 75 | 1 800.00 | 1 | 0.003 960 | 7.1280 | 0.003 960 |
| 58 | 300 | 0 | 150 | 500 | 450 | 50 | 1 800.00 | 1 | 0.003 960 | 7.1280 | 0.003 960 |
| 59 | 300 | 0 | 150 | 600 | 510 | 90 | 1 800.00 | 1 | 0.001 980 | 3.5640 | 0.001 980 |
| 60 | 300 | 0 | 150 | 600 | 540 | 60 | 1 800.00 | 1 | 0.001 980 | 3.5640 | 0.001 980 |
| 61 | 300 | 200 | 0 | 200 | 170 | 30 | 0 | 0 | 0.017 820 | 0 | 0 |
| 62 | 300 | 200 | 0 | 200 | 180 | 20 | 0 | 0 | 0.017 820 | 0 | 0 |
| 63 | 300 | 200 | 0 | 300 | 255 | 45 | 0 | 0 | 0.035 640 | 0 | 0 |
| 64 | 300 | 200 | 0 | 300 | 270 | 30 | 0 | 0 | 0.035 640 | 0 | 0 |
| 65 | 300 | 200 | 0 | 400 | 340 | 60 | 542.05 | 0 | 0.022 275 | 12.0742 | 0 |
| 66 | 300 | 200 | 0 | 400 | 360 | 40 | 645.00 | 0 | 0.022 275 | 14.0742 | 0 |
| 67 | 300 | 200 | 0 | 500 | 425 | 75 | 1 537.81 | 0 | 0.008 910 | 13.7019 | 0 |
| 68 | 300 | 200 | 0 | 500 | 450 | 50 | 1 545.00 | 0 | 0.008 910 | 13.7660 | 0 |

Table B.3 Enumeration of scenarios in the Akabuga District simulation.

| Scenario | \bar{P} | \bar{G}_1 | \bar{G}_2 | D_{tot} | D_1 | D_2 | TOC | LOLO | f_Y | $f_Y TOC$ | $f_Y LOLO$ |
|----------|-----------|-------------|-------------|-----------|-------|-------|----------|------|-----------|-----------|------------|
| 69 | 300 | 200 | 0 | 600 | 510 | 90 | 2 000.00 | 1 | 0.004 455 | 8.9100 | 0.004 455 |
| 70 | 300 | 200 | 0 | 600 | 540 | 60 | 2 000.00 | 1 | 0.004 455 | 8.9100 | 0.004 455 |
| 71 | 300 | 200 | 150 | 200 | 170 | 30 | 0 | 0 | 0.071 280 | 0 | 0 |
| 72 | 300 | 200 | 150 | 200 | 180 | 20 | 0 | 0 | 0.071 280 | 0 | 0 |
| 73 | 300 | 200 | 150 | 300 | 255 | 45 | 0 | 0 | 0.142 560 | 0 | 0 |
| 74 | 300 | 200 | 150 | 300 | 270 | 30 | 0 | 0 | 0.142 560 | 0 | 0 |
| 75 | 300 | 200 | 150 | 400 | 340 | 60 | 542.05 | 0 | 0.089 100 | 48.2967 | 0 |
| 76 | 300 | 200 | 150 | 400 | 360 | 40 | 645.00 | 0 | 0.089 100 | 57.4695 | 0 |
| 77 | 300 | 200 | 150 | 500 | 425 | 75 | 1 537.81 | 0 | 0.035 640 | 54.8076 | 0 |
| 78 | 300 | 200 | 150 | 500 | 450 | 50 | 1 545.00 | 0 | 0.035 640 | 55.0638 | 0 |
| 79 | 300 | 200 | 150 | 600 | 510 | 90 | 2 640.56 | 0 | 0.017 820 | 47.0548 | 0 |
| 80 | 300 | 200 | 150 | 600 | 540 | 60 | 2 650.46 | 0 | 0.017 820 | 47.2312 | 0 |

Simulation Results

Table shows the results of all tests of the Akabuga District simulation.

To be added: Comments...

Table B.4 Compilation of simulation results for Akabuga District.

| Simulation method | Average simulation time [ms] | M_{TOC} | | | | | M_{LOLO} | | | | |
|--|------------------------------|-----------|-------|-------|-------|-------|------------|--------|--------|----------|----------|
| | | Min. | Mean | Max. | Var. | Eff. | Min. | Mean | Max. | Var. | Eff. |
| Enumeration | 27 | | 443.4 | | 0 | 0 | | 0.0331 | | 0 | 0 |
| Simple sampling | 8 | 366.1 | 436.0 | 488.8 | 480.7 | 3 758 | 0.0200 | 0.0326 | 0.0480 | 0.000034 | 0.000263 |
| Complementary random numbers: $(G, D), (G^*, D^*)$ | 11 | 391.9 | 444.0 | 484.8 | 311.2 | 3 303 | 0.0160 | 0.0332 | 0.0520 | 0.000036 | 0.000384 |
| Complementary random numbers: $(G, D), (G, D^*), (G, D^*), (G^*, D^*)$ | 8 | 388.2 | 441.9 | 491.6 | 522.6 | 4 120 | 0.0210 | 0.0331 | 0.0450 | 0.000034 | 0.000266 |
| Dagger sampling | 11 | 408.3 | 444.7 | 509.0 | 438.7 | 4 995 | 0.0210 | 0.0334 | 0.0480 | 0.000029 | 0.000327 |
| Control variates | 13 | 424.7 | 442.2 | 467.4 | 53.4 | 688 | 0.0250 | 0.0327 | 0.0410 | 0.000012 | 0.000151 |
| Importance sampling | 8 | 416.1 | 440.1 | 474.6 | 155.6 | 1 292 | 0.0262 | 0.0331 | 0.0435 | 0.000013 | 0.000104 |
| Stratified sampling | 12 | 417.9 | 443.2 | 476.1 | 134.2 | 1 578 | 0.0330 | 0.0331 | 0.0332 | 0.000000 | 0.000000 |

B.2 Simulation Code

Below follows the Matlab code used for simulation of Akabuga District. Notice that the code does not always follows a good programming style; for example, calculations that are carried out in several of the simulations (such as applying the inverse transform method to generate input values) could have been written as separate subroutines (i.e., functions in Matlab). However, in these tests computation time is of great importance, and scripts are faster than functions. The only exception is the mathematical model, where at least the detailed mathematical model will be called the same number of times in all simulations.

akabugadistrictscenario.m

This is the core of the simulation: the detailed mathematical model of the system. It takes a scenario as inputs and computes the total operation cost as well as the binary loss of load occasion variable.

```
function [toc,lolo] = akabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2)
%AKABUGADISTRICTSCENARIO Calculates samples for a scenario in Akabuga District.
```

```

%AKABUGADISTRICTSCENARIO Calculates samples for a scenario in Akabuga District.
% [TOC,LOLO] = MJIREGIONSCENARIO(G1MAX,G2MAX,PMAX,DTOT,D1,D2) calculates
% operation cost and loss of load for a scenario in Akabuga District.
% G1MAX and % G2MAX are the available capacities in the diesel generator
% sets, PMAX is the available transmission capacity, DTOT, D1 and D2 are
% the total load, load in Akabuga and load in Ekyaaloo respectively.

% Calculate possible export from Ekyaaloo
Pexp = min(Pmax,350-D2);

% Calculate generation in Akabuga
Pimp = Pexp - 5e-5*Pexp^2;
if D1 <= Pimp
    % Load in Akabuga can be covered by import from Ekyaaloo
    toc = 0; loloo = 0; return
else
    U1 = D1 - Pimp;
end

if U1 <= G1max
    % Load in Akabuga can be covered by import and the larger diesel
    % generator set
    toc = 10*U1; loloo = 0; return
else
    U1 = U1 - G1max;
end

if U1 <= G2max
    % Load in Akabuga can be covered by import and both diesel
    % generator sets
    toc = 10*G1max + 12*U1; loloo = 0; return
else
    % Load shedding
    toc = 10*G1max + 12*G2max; loloo = 1; return
end

```

akabugadistrictsimplesampling.m

This script executes one simulation using simple sampling. The principle is straightforward: generate a scenario by applying the inverse transform method on each input, analyse the scenario using the mathematical model and store the results.

```

% Simple sampling of Akabuga District
tocsum = 0; lolosum = 0;
for i = 1:n
    % Randomise scenario
    u = rand;
    if u <= .9
        G1max = 200;
    else
        G1max = 0;
    end
    u = rand;
    if u <= .8
        G2max = 150;
    else
        G2max = 0;
    end
    u = rand;
    if u <= .99
        Pmax = 300;
    else
        Pmax = 0;
    end
    u = rand;

```

```

if u < .2
    Dtot = 200;
elseif u < .6
    Dtot = 300;
elseif u < .85
    Dtot = 400;
elseif u < .95
    Dtot = 500;
else
    Dtot = 600;
end
u = rand;
if u <= .5
    c = .85;
else
    c = .9;
end
D1 = c*Dtot; D2 = (1-c)*Dtot;
% Analyse scenario
[toci,loloi] = akabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2);
% Store results
tocsum = tocsum + toci;
lolosum = lolosum + loloi;
end
% Compute estimates
mTOC = tocsum/n;
mLOLO = lolosum/n;

```

akabugadistrictcomprandnumb2.m

This script executes one simulation using complementary random numbers. In this variant each original scenario, (G_{tot}, D) , generates one complementary scenario, (G_{tot}^*, D^*) . The principle is the same as for simple sampling, except for the additional code that switches between generating original and complementary random numbers. However, as we can see in table B.4, this tiny extra code increases the average simulation time by approximately 35%!

```

% Simulation of Akabuga District using complementary random numbers for total
% load and total generation capacity (1 complementary scenario)
tocsum = 0; lolosum = 0; cycle = 1;
for i = 1:n
    % Randomise original scenario
    if cycle == 1
        uG = rand; uD = rand; cycle = 2;
    else
        uG = 1 - uG; uD = 1 - uD; cycle = 1;
    end
    if uG > .28
        G1max = 200; G2max = 150;
    elseif uG > .1
        G1max = 200; G2max = 0;
    elseif uG > .02
        G1max = 0; G2max = 150;
    else
        G1max = 0; G2max = 0;
    end
    u = rand;
    if u <= .99
        Pmax = 300;
    else
        Pmax = 0;
    end
    if uD < .2
        Dtot = 200;
    elseif uD < .6
        Dtot = 300;

```

```

elseif uD < .85
    Dtot = 400;
elseif uD < .95
    Dtot = 500;
else
    Dtot = 600;
end
u = rand;
if u <= .5
    c = .85;
else
    c = .9;
end
D1 = c*Dtot; D2 = (1-c)*Dtot;
% Analyse original scenario
[toci,loloi] = akabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2);
% Store results
tocsum = tocsum + toci;
lolosum = lolosum + loloi;
end
% Compute estimates
mTOC = tocsum/n;
mLOLO = lolosum/n;

```

akabugadistrictcomplrandnumb4.m

This script executes one simulation using complementary random numbers. In this variant each original scenario, (G_{tot}, D) , generates three complementary scenarios, (G_{tot}^*, D^*) , (G_{tot}^*, D^*) and (G_{tot}^*, D^*) respectively. The principle is the same as for simple sampling, except for the additional code that switches between generating original and complementary random numbers. This variant require only 75% less pseudorandom numbers to generate the values of the generation capacity and the total load, which compensates for the extra time to switch between generating original and complementary random numbers; therefore the average simulation time is more or less the same as for simple sampling.

```

% Simulation of Akabuga District using complementary random numbers for total
% load and total generation capacity (3 complementary scenario)
tocsum = 0; lolosum = 0; cycle = 1;
for i = 1:n
    % Randomise original scenario
    if cycle == 1
        uG = rand; uD = rand; cycle = 2;
    elseif cycle == 2
        uD = 1 - uD; cycle = 3;
    elseif cycle == 3
        uG = 1 - uG; cycle = 4;
    else
        uD = 1 - uD; cycle = 1;
    end
    if uG > .28
        G1max = 200; G2max = 150;
    elseif uG > .1
        G1max = 200; G2max = 0;
    elseif uG > .02
        G1max = 0; G2max = 150;
    else
        G1max = 0; G2max = 0;
    end
    u = rand;
    if u <= .99
        Pmax = 300;
    else
        Pmax = 0;
    end
    if uD < .2

```

```

        Dtot = 200;
elseif uD < .6
        Dtot = 300;
elseif uD < .85
        Dtot = 400;
elseif uD < .95
        Dtot = 500;
else
        Dtot = 600;
end
u = rand;
if u <= .5
        c = .85;
else
        c = .9;
end
D1 = c*Dtot; D2 = (1-c)*Dtot;
% Analyse original scenario
[toci,loloi] = akabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2);
% Store results
tocsum = tocsum + toci;
lolosum = lolosum + loloi;
end
% Compute estimates
mTOC = tocsum/n;
mLOLO = lolosum/n;

```

akabugadistrictcontrolvariates.m

This script executes one simulation using control variates. First, a probabilistic production cost simulation is carried out using the *akabugadistrictppc* script (see below). Then, simple sampling is applied to the difference between the output of the detailed model of the Akabuga district and the simplified model. Finally, the expectation values of the control variates are added to the estimated differences.

```

% Simulation of Akabuga District using control variates
akabugadistrictppc
tocdiffsum = 0; lolodiffsum = 0;
for i = 1:n
    % Randomise scenario
    u = rand;
    if u <= .9
        G1max = 200;
    else
        G1max = 0;
    end
    u = rand;
    if u <= .8
        G2max = 150;
    else
        G2max = 0;
    end
    u = rand;
    if u <= .99
        Pmax = 300;
    else
        Pmax = 0;
    end
    u = rand;
    if u < .2
        Dtot = 200;
    elseif u < .6
        Dtot = 300;
    elseif u < .85

```



```

        Dtot = 400;
elseif u < .95
        Dtot = 500;
else
        Dtot = 600;
end
u = rand;
if u <= .5
        c = .85;
else
        c = .9;
end
D1 = c*Dtot; D2 = (1-c)*Dtot;
% Analyse scenario
[toci,loloi] = akabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2);
[toctilde,lolotilde] = ...
        simpleakabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2);
% Store results
tocdiffsum = tocdiffsum + toci - toctilde;
lolodiffsum = lolodiffsum + loloi - lolotilde;
end
% Compute estimates
mTOC = tocdiffsum/n + ETOCppc;
mLOLO = lolodiffsum/n + LOLPppc;

```

akabugadistrictppc.m

This script applies probabilistic production cost simulation (see appendix C) to the Akabuga district. The equivalent load duration curves (F1, F2 and F3) are stored as row vectors, where the first element represents the value of the duration curve for $0 \leq x < 50$, the second element is for $50 \leq x < 100$, etc.

```

% Probabilistic production cost simulation of Akabuga District
F1 = [1 1 1 1 .8 .8 .4 .4 .15 .15 .05 .05 0 0 0 0 0 0 0];
F2 = .9*F1 + .1*[1 1 1 1 F1(1:end-4)];
F3 = .8*F2 + .2*[1 1 1 F2(1:end-3)];
EENS1 = 0;
for i = (350/50):19
        EENS1 = EENS1 + 50*F1(i+1);
end
EENS2 = 0;
for i = (550/50):19
        EENS2 = EENS2 + 50*F2(i+1);
end
EENS3 = 0;
for i = (700/50):19
        EENS3 = EENS3 + 50*F3(i+1);
end
ETOCppc = 10*(EENS1-EENS2) + 12*(EENS2 - EENS3);
LOLPppc = F3(700/50+1);

```

simpleakabugadistrictscenario.m

This is the simplified mathematical model. It takes a scenario as inputs and computes the total operation cost as well as the binary loss of load occasion variable.

```

function [toc,lolo] = simpleakabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2)
%SIMPLEAKABUGADISTRICTSCENARIO Calculates samples for a scenario in Akabuga
%
%                               District using a simplified model.
%   [TOC,LOLO] = SIMPLEAKABUGADISTRICTSCENARIO(G1MAX,G2MAX,PMAX,DTOT,D1,D2)
%   calculates % operation cost and loss of load for a scenario in Akabuga
%   District. G1MAX and G2MAX are the available capacities in the diesel
%   generator sets, PMAX is the available transmission capacity, DTOT, D1 and
%   D2 are the total load, load in Akibuga and load in Ekyaaloo respectively.

if Dtot <= 350

```

```

    % Total load can be covered by hydro power
    toc = 0; lololo = 0; return
else
    U = Dtot - 350;
end

if U <= G1max
    % Total load can be covered by hydro power and the larger diesel
    % generator set
    toc = 10*U; lololo = 0; return
else
    U = U - G1max;
end
if U <= G2max
    % Total load can be covered by hydro power and both diesel
    % generator sets
    toc = 10*G1max + 12*U; lololo = 0; return
else
    % Load shedding
    toc = 10*G1max + 12*G2max; lololo = 1; return
end

```

akabugadistrictimportancesampling.m

This script executes one simulation using importance sampling. The principle is mostly the same as for simple sampling, except that scenarios are randomised according to the importance sampling functions and that each observation is multiplied by a weight factor. The weight factors for each input are assigned at the same time as the random numbers are computed. For example, if the large generator is available, the weight factor is $w_{G1} = f_{G1}(200)/f_{Z1}(200) = 0.9/0.8 = 1.125$.

```

% Importance sampling of Akabuga District
tocsum = 0; lolosum = 0;
for i = 1:n
    % Randomise scenario
    u = rand;
    if u <= .8
        G1max = 200; wG1max = 1.125;
    else
        G1max = 0; wG1max = .5;
    end
    u = rand;
    if u <= .8
        G2max = 150; wG2max = 1;
    else
        G2max = 0; wG2max = 1;
    end
    u = rand;
    if u <= .98
        Pmax = 300; wPmax = .99/.98;
    else
        Pmax = 0; wPmax = .5;
    end
    u = rand;
    if u <= .2
        Dtot = 200; wDtot = 1;
    elseif u <= .4
        Dtot = 300; wDtot = 2;
    elseif u <= .6
        Dtot = 400; wDtot = 1.25;
    elseif u <= .8
        Dtot = 500; wDtot = .5;
    else
        Dtot = 600; wDtot = .25;
    end
end

```

```

u = rand;
if u <= .5
    c = .85;
else
    c = .9;
end
D1 = c*Dtot; D2 = (1-c)*Dtot;
% Analyse scenario
[toci,loloi] = akabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2);
% Store results
w = wG1max*wG2max*wPmax*wDtot;
tocsum = tocsum + w*toci;
lolosum = lolosum + w*loloi;
end
% Compute estimates
mTOC = tocsum/n;
mLOLO = lolosum/n;

```

akabugadistrictstratifiedsampling.m

This script executes one simulation using importance sampling. First, the stratum weights are computed and analytical results are assigned to those strata where it is possible. The next step is to run a pilot study; the number of samples are the same for all strata for which there is no analytical result. The randomisation of the input values is modified to take into account the probability distribution of the inputs for the corresponding strata. Consider for example the randomisation of the total load, which starts by checking if the load is fixed (i.e., only has one possible outcome) for the stratum in question. If that is not the case, the pseudorandom number u is scaled to fit the possible range of total load values for the stratum and then the inverse transform method is applied to the scaled pseudorandom number (cf. example 5.6).

After the pilot study, estimates of the variance for all sampled strata are calculated and the sample allocation for the next batch of scenarios is computed according to algorithm 5.3. Scenarios are then generated in the same way as in the pilot study. The results for each stratum are stored and a new batch is generated according to an updated sample allocation. At the end of the simulation, the results are weighted together according to the stratum weights.

```

% Stratified sampling of Akabuga District

% Define strata
L = 14;
omega = [.01*.02 ...
         .01*.08 ...
         .01*.18*.2 ...
         .01*.18*.8 ...
         .01*.72*.6 ... % Stratum 5
         .01*.72*.25 ...
         .01*.72*.15 ...
         .99*.02*.6 + .99*.08*.6 + .99*.18*.6 + .99*.72*.6 ...
         .99*.02*.4 ...
         .99*.08*.25 ... % Stratum 10
         .99*.08*.15 ...
         .99*.18*.35 ...
         .99*.18*.05 ...
         .99*.72*.4];

mTOCh = zeros(1,L); mLOLOh = zeros(1,L);
sTOCh = zeros(1,L); sLOLOh = zeros(1,L);
muTOCh = NaN(1,L); muLOLOh = NaN(1,L);
muTOCh(1) = 0; muLOLOh(1) = 1;
muTOCh(2) = 1800; muLOLOh(2) = 1;
muTOCh(4) = 2000; muLOLOh(4) = 1;
muTOCh(7) = 3800; muLOLOh(7) = 1;
muTOCh(8) = 0; muLOLOh(8) = 0;
muTOCh(9) = 0; muLOLOh(9) = 1;
muTOCh(11) = 1800; muLOLOh(11) = 1;

```

```

muTOCh(13) = 2000; muLOLOh(13) = 1;
nh = zeros(1,L);
tocsumh = zeros(1,L); lolosumh = zeros(1,L);
toc2sumh = zeros(1,L); lolo2sumh = zeros(1,L);

% Pilot study
nhpilot = 25;
for h = 1:L
    if isnan(muTOCh(h))
        for i = 1:nhpilot
            if h == 3 || h == 12
                G1max = 200; G2max = 0;
            elseif h == 5 || h == 6 || h == 14
                G1max = 200; G2max = 150;
            elseif h == 10
                G1max = 0; G2max = 150;
            end
            if h <= 7
                Pmax = 0;
            else
                Pmax = 300;
            end
            if h == 3
                Dtot = 200;
            elseif h == 6 || h == 10
                Dtot = 400;
            else
                u = rand;
                if h == 4
                    u = .8*u + .2;
                elseif h == 5 || h == 8
                    u = .6*u;
                elseif h == 12
                    u = .35*u + .6;
                elseif h == 14
                    u = .4*u + .6;
                end
                if u < .2
                    Dtot = 200;
                elseif u < .6
                    Dtot = 300;
                elseif u < .85
                    Dtot = 400;
                elseif u < .95
                    Dtot = 500;
                else
                    Dtot = 600;
                end
            end
            u = rand;
            if u <= .5
                c = .85;
            else
                c = .9;
            end
            D1 = c*Dtot; D2 = (1-c)*Dtot;
            % Analyse scenario
            [toci,loloi] = ...
                akabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2);
            % Store results
            tocsumh(h) = tocsumh(h) + toci;
            toc2sumh(h) = toc2sumh(h) + toci^2;
            lolosumh(h) = lolosumh(h) + loloi;
            lolo2sumh(h) = lolo2sumh(h) + loloi^2;
        end
    end
end

```

```

end
nh(h) = nhpilot;
mTOCh(h) = tocsumh(h)/nh(h);
sTOCh(h) = toc2sumh(h)/nh(h) - (mTOCh(h))^2;
mLOLOh(h) = lolosumh(h)/nh(h);
sLOLOh(h) = lolos2sumh(h)/nh(h) - (mLOLOh(h))^2;
else
mTOCh(h) = muTOCh(h);
mLOLOh(h) = muLOLOh(h);
end
end
npilot = sum(nh);

% Simulation
b = 0;
while sum(nh) < n
b = b + 1;
% Determine Neyman allocation
ntarget = npilot + 50*b;
nhtarget = (ntarget*omega.*sTOCh/(omega*sTOCh') + ...
ntarget*omega.*sLOLOh/(omega*sLOLOh'))/2;
nhbtarget = nhtarget - nh;
Hplus = nhbtarget > 0; nplus = sum(nhbtarget(Hplus));
Hminus = nhbtarget < 0; nminus = -sum(nhbtarget(Hminus));
nhb = zeros(1,L);
nhb(Hplus) = round((1-nminus/nplus)*nhbtarget(Hplus));
nh = nh + nhb;
for h = 1:L
if isnan(muTOCh(h))
for i = 1:nhb(h)
if h == 3 || h == 12
G1max = 200; G2max = 0;
elseif h == 5 || h == 6 || h == 14
G1max = 200; G2max = 150;
elseif h == 10
G1max = 0; G2max = 150;
end
if h <= 7
Pmax = 0;
else
Pmax = 300;
end
if h == 3
Dtot = 200;
elseif h == 6 || h == 10
Dtot = 400;
else
u = rand;
if h == 4
u = .8*u + .2;
elseif h == 5 || h == 8
u = .6*u;
elseif h == 12
u = .35*u + .6;
elseif h == 14
u = .4*u + .6;
end
if u < .2
Dtot = 200;
elseif u < .6
Dtot = 300;
elseif u < .85
Dtot = 400;
elseif u < .95

```

```

        Dtot = 500;
    else
        Dtot = 600;
    end
end
u = rand;
if u <= .5
    c = .85;
else
    c = .9;
end
D1 = c*Dtot; D2 = (1-c)*Dtot;
% Analyse scenario
[toci,loloi] = ...
    akabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2);
% Store results
tocsumh(h) = tocsumh(h) + toci;
toc2sumh(h) = toc2sumh(h) + toci^2;
lolosumh(h) = lolosumh(h) + loloi;
lolo2sumh(h) = lolo2sumh(h) + loloi^2;
end
mTOCh(h) = tocsumh(h)/nh(h);
sTOCh(h) = toc2sumh(h)/nh(h) - (mTOCh(h))^2;
mLOLOh(h) = lolosumh(h)/nh(h);
sLOLOh(h) = lolo2sumh(h)/nh(h) - (mLOLOh(h))^2;
end
end
end

% Compile final results
mTOC = omega*mTOCh';
mLOLO = omega*mLOLOh';

```

akabugadistrictevaluation.m

This script is used to test and compare the results of different simulation method. First, the true expectation values of the outputs are computed using enumeration. Then the same number of samples is collected in each simulation, and number of tests are the same for all methods. Moreover, each test run uses the same seed for the random number generator for all methods. This means that two methods that generate scenarios in exactly the same way (such as simple sampling and control variates) will use the same scenarios in each test run.

The time of each simulation run is measured using the built-in Matlab timer (`tic` and `toc`). Notice that the execution time will vary slightly every time this script is run, depending on which other tasks are running on the same computer, how much memory that is available, etc.

The last part of the script presents the results of the tests on the screen.

```

clear
akabugadistrictenumeration

% Evaluation of simulation methods for Akabuga District
n = 1000; % Number of samples per simulation
ntest = 100; % Number of tests of each simulation method
simplesamplingres = zeros(3,ntest);
comprandnumbres2 = zeros(3,ntest);
comprandnumbres4 = zeros(3,ntest);
daggersamplingres = zeros(3,ntest);
controlvariatesres = zeros(3,ntest);
importancesamplingres = zeros(3,ntest);
stratifiedsamplingres = zeros(3,ntest);

for test = 1:ntest
    rng(test,'twister'); % Seed for random number generator
    tstart = tic;
    akabugadistrictsimplesampling

```

```

simplerandomres(3,test) = toc(tstart);
simplerandomres(1,test) = mTOC;
simplerandomres(2,test) = mLLOLO;
rng(test,'twister'); % Seed for random number generator
tstart = tic;
akabugadistrictcomprandnumb2
comprandnumb2(3,test) = toc(tstart);
comprandnumb2(1,test) = mTOC;
comprandnumb2(2,test) = mLLOLO;
rng(test,'twister'); % Seed for random number generator
tstart = tic;
akabugadistrictcomprandnumb4
comprandnumb4(3,test) = toc(tstart);
comprandnumb4(1,test) = mTOC;
comprandnumb4(2,test) = mLLOLO;
rng(test,'twister'); % Seed for random number generator
tstart = tic;
akabugadistrictdaggersampling
daggersamplingres(3,test) = toc(tstart);
daggersamplingres(1,test) = mTOC;
daggersamplingres(2,test) = mLLOLO;
rng(test,'twister'); % Seed for random number generator
tstart = tic;
akabugadistrictcontrolvariates
controlvariatesres(3,test) = toc(tstart);
controlvariatesres(1,test) = mTOC;
controlvariatesres(2,test) = mLLOLO;
rng(test,'twister'); % Seed for random number generator
tstart = tic;
akabugadistrictimportancesampling
importancesamplingres(3,test) = toc(tstart);
importancesamplingres(1,test) = mTOC;
importancesamplingres(2,test) = mLLOLO;
rng(test,'twister'); % Seed for random number generator
tstart = tic;
akabugadistrictstratifiedsampling
stratifiedsamplingres(3,test) = toc(tstart);
stratifiedsamplingres(1,test) = mTOC;
stratifiedsamplingres(2,test) = mLLOLO;
end

avsimtime = 1000*mean(simplerandomres(3,:));
fprintf('Simple sampling (Average simulation time: %.1f ms)\n', avsimtime);
tocres = simplerandomres(1,:);
fprintf(' TOC - Min:% 8.1f Mean:% 8.1f Max:% 8.1f', ...
        min(tocres),mean(tocres),max(tocres));
fprintf(' Var: % 8.3f Eff: %f\n', var(tocres), avsimtime*var(tocres));
lolores = simplerandomres(2,:);
fprintf(' LOLO - Min:% 8.4f Mean:% 8.4f Max:% 8.4f', ...
        min(lolores),mean(lolores),max(lolores));
fprintf(' Var: %.6f Eff: %f\n', var(lolores), avsimtime*var(lolores));

avsimtime = 1000*mean(comprandnumb2(3,:));
fprintf(['Complementary random numbers with 1 complementary scenario ' ...
        '(Average simulation time: %.1f ms)\n'], avsimtime);
tocres = comprandnumb2(1,:);
fprintf(' TOC - Min:% 8.1f Mean:% 8.1f Max:% 8.1f', ...
        min(tocres),mean(tocres),max(tocres));
fprintf(' Var: % 8.3f Eff: %f\n', var(tocres), avsimtime*var(tocres));
lolores = comprandnumb2(2,:);
fprintf(' LOLO - Min:% 8.4f Mean:% 8.4f Max:% 8.4f', ...
        min(lolores),mean(lolores),max(lolores));
fprintf(' Var: %.6f Eff: %f\n', var(lolores), avsimtime*var(lolores));

```

```

avsimtime = 1000*mean(complrandnombres4(3,:));
fprintf(['Complementary random numbers with 3 complementary scenario ' ...
'(Average simulation time: %.1f ms)\n'], avsimtime);
tocres = complrandnombres4(1,:);
fprintf(' TOC - Min:% 8.1f Mean:% 8.1f Max:% 8.1f', ...
min(tocres),mean(tocres),max(tocres));
fprintf(' Var: % 8.3f Eff: %f\n', var(tocres), avsimtime*var(tocres));
lolores = complrandnombres4(2,:);
fprintf(' LOLO - Min:% 8.4f Mean:% 8.4f Max:% 8.4f', ...
min(lolores),mean(lolores),max(lolores));
fprintf(' Var: %.6f Eff: %f\n', var(lolores), avsimtime*var(lolores));

avsimtime = 1000*mean(daggersamplingres(3,:));
fprintf('Dagger sampling (Average simulation time: %.1f ms)\n', avsimtime);
tocres = daggersamplingres(1,:);
fprintf(' TOC - Min:% 8.1f Mean:% 8.1f Max:% 8.1f', ...
min(tocres),mean(tocres),max(tocres));
fprintf(' Var: % 8.3f Eff: %f\n', var(tocres), avsimtime*var(tocres));
lolores = daggersamplingres(2,:);
fprintf(' LOLO - Min:% 8.4f Mean:% 8.4f Max:% 8.4f', ...
min(lolores),mean(lolores),max(lolores));
fprintf(' Var: %.6f Eff: %f\n', var(lolores), avsimtime*var(lolores));

avsimtime = 1000*mean(controlvariatesres(3,:));
fprintf('Control variates (Average simulation time: %.1f ms)\n', avsimtime);
tocres = controlvariatesres(1,:);
fprintf(' TOC - Min:% 8.1f Mean:% 8.1f Max:% 8.1f', ...
min(tocres),mean(tocres),max(tocres));
fprintf(' Var: % 8.3f Eff: %f\n', var(tocres), avsimtime*var(tocres));
lolores = controlvariatesres(2,:);
fprintf(' LOLO - Min:% 8.4f Mean:% 8.4f Max:% 8.4f', ...
min(lolores),mean(lolores),max(lolores));
fprintf(' Var: %.6f Eff: %f\n', var(lolores), avsimtime*var(lolores));

avsimtime = 1000*mean(importancesamplingres(3,:));
fprintf('Importance sampling (Average simulation time: %.1f ms)\n', avsimtime);
tocres = importancesamplingres(1,:);
fprintf(' TOC - Min:% 8.1f Mean:% 8.1f Max:% 8.1f', ...
min(tocres),mean(tocres),max(tocres));
fprintf(' Var: % 8.3f Eff: %f\n', var(tocres), avsimtime*var(tocres));
lolores = importancesamplingres(2,:);
fprintf(' LOLO - Min:% 8.4f Mean:% 8.4f Max:% 8.4f', ...
min(lolores),mean(lolores),max(lolores));
fprintf(' Var: %.6f Eff: %f\n', var(lolores), avsimtime*var(lolores));

avsimtime = 1000*mean(stratifiedsamplingres(3,:));
fprintf('Stratified sampling (Average simulation time: %.1f ms)\n', avsimtime);
tocres = stratifiedsamplingres(1,:);
fprintf(' TOC - Min:% 8.1f Mean:% 8.1f Max:% 8.1f', ...
min(tocres),mean(tocres),max(tocres));
fprintf(' Var: % 8.3f Eff: %f\n', var(tocres), avsimtime*var(tocres));
lolores = stratifiedsamplingres(2,:);
fprintf(' LOLO - Min:% 8.4f Mean:% 8.4f Max:% 8.4f', ...
min(lolores),mean(lolores),max(lolores));
fprintf(' Var: %.6f Eff: %f\n', var(lolores), avsimtime*var(lolores));

```

akabugadistrictenumeration.m

This script computes the expectation values of a simulation of Akabuga District using enumeration. The possible states of the inputs and the probability of each state are stored in the matrices *Y* and *fY* respectively. The script then loops through all possible scenarios and calculates the output of the scenario as well as the probability of the scenario. The results are computed according to the definition of expectation value for discrete probability distributions and the results are presented on the screen.


```

% Computation of true expectation values for simulation of Akabuga District

tstart = tic;
Y = zeros(5,5); fY = zeros(size(Y));
Y(1,1:2) = [0 200]; fY(1,1:2) = [.1 .9];
Y(2,1:2) = [0 150]; fY(2,1:2) = [.2 .8];
Y(3,1:2) = [0 300]; fY(3,1:2) = [.01 .99];
Y(4,:) = 200:100:600; fY(4,:) = [.2 .4 .25 .1 .05];
Y(5,1:2) = [.85 .9]; fY(5,1:2) = [.5 .5];
muTOC = 0; muTOCtilde = 0;
muLOLO = 0; muLOLOtilde = 0;

for i1 = 1:2
    for i2 = 1:2
        for i3 = 1:2
            for i4 = 1:5
                for i5 = 1:2
                    G1max = Y(1,i1);
                    G2max = Y(2,i2);
                    Pmax = Y(3,i3);
                    Dtot = Y(4,i4);
                    c = Y(5,i5);
                    D1 = c*Dtot; D2 = (1-c)*Dtot;
                    [toci,loloi] = ...
                        akabugadistrictscenario(G1max,G2max,Pmax,Dtot,D1,D2);
                    fYi = fY(1,i1)*fY(2,i2)*fY(3,i3)*fY(4,i4)*fY(5,i5);
                    muTOC = muTOC + fYi*toci;
                    muLOLO = muLOLO + fYi*loloi;
                end
            end
        end
    end
end
comptime = toc(tstart);

fprintf('Enumeration (Computation time: %.1f ms)\n', 1000*comptime);
fprintf(' TOC - % 8.1f\n', muTOC);
fprintf(' LOLO - % 8.4f\n', muLOLO);

```


PROBABILISTIC PRODUCTION COST SIMULATION

This appendix provides a short overview of probabilistic production cost simulation, which is an analytical model for calculation of the operation cost and reliability of power systems. The model used in this simulation method is rather limited, but it can be used to generate control variates for more detailed electricity market simulation (such as in the example of Akabuga District). Further details on probabilistic production cost simulation can be found in the literature listed at the end of the appendix.

Further reading

- H. Baleriaux, E. Jamouille & F. Linard de Guertechin, “Simulation de l’explittion d’un parc de machines thermiques de production d’électricité couplé à des stations de pompage”, *Extrait de la revue E* (édition S.R.B.E), Vol. 5, No. 7, 1967.
- R. R. Booth, “Power System Simulation Model Based on Probability Analysis”, *IEEE Transactions on Power Apparatus & Systems*, Vol. PAS-91, No. 1, January/February 1972.
- L. Söder & M. Amelin, “Efficient Operation and Planning of Power Systems”, course compendium, Royal Institute of Technology (KTH), Stockholm 2003–2015.

SOLUTIONS TO EXERCISES

2.1

a) All values between 10 and 20 are equally probable; thus, this is a heterogeneous population.

b) The values 10, 11, ..., 20 are equally probable; thus, this is a heterogeneous population.

c) Only the values 10 and 20 are possible; however, the value 10 has a probability of 60%. Hence, this random variable corresponds to a population where 60% of the units have the value 10 and the remainder of the units correspond to the value 20. Hence, this is a duogeneous population, where the units with the value 10 are the conformist units and the other units are the diverging units.

d) All values between 10 and 20 are possible; however, the value 10 has a probability of 60%. Hence, this random variable corresponds to a population where 60% of the units have the value 10 and the remainder of the units correspond to all values larger than 10 and smaller or equal to 20. Hence, this is a duogeneous population, where the units with the value 10 are the conformist units and the other units are the diverging units.

e) All values between 10 and 20 are possible and although values between 10 and 12 are more probable there is no particular value that is dominating, which means that we cannot identify a set of conformist units. Hence, this is a heterogeneous population.

3.1

a) $y = F_Y^{-1}(0.16) = 4.$

b) $y = F_Y^{-1}(0.28) = 4.$

c) $y = F_Y^{-1}(0.55) = 6.$

d) $y = F_Y^{-1}(0.96) = 10.$

3.2

a) $y = F_Y^{-1}(0.10) = 2.$

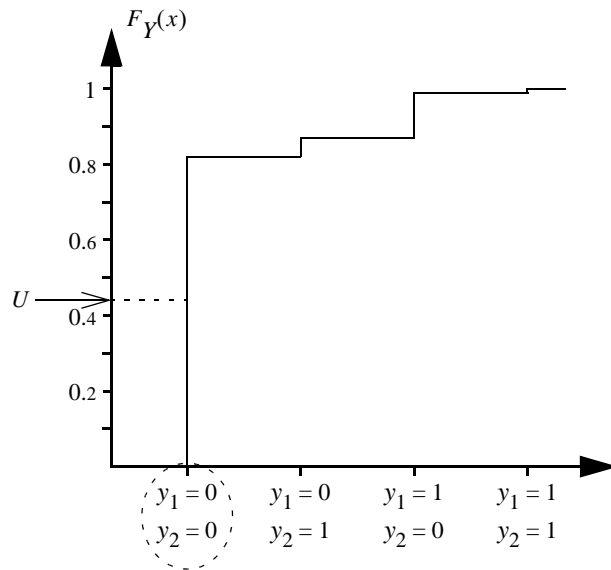
b) $y = F_Y^{-1}(0.25) = 3.25.$

c) $y = F_Y^{-1}(0.63) = 5.15.$

d) $y = F_Y^{-1}(0.91) = 7.1.$

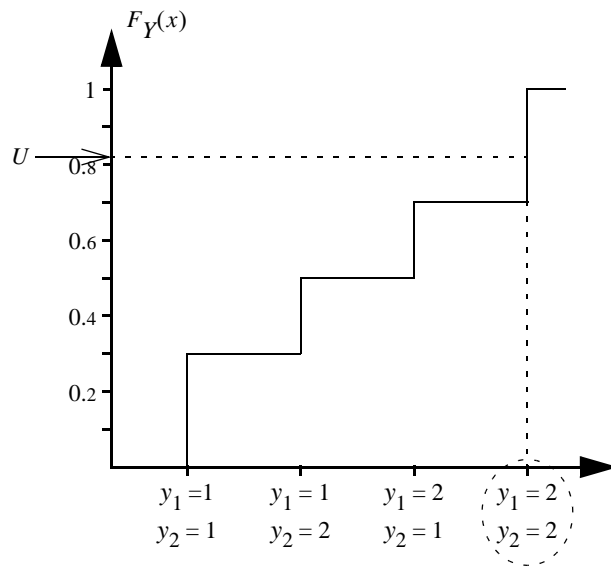
3.5

$F_Y^{-1}(0.46)$ can be calculated graphically as indicated in the figure below. The result is $y_1 = 0$ and $y_2 = 0$.



3.4

The inverse transform method can be applied to the joint distribution function, as shown in the figure below. Using the first of the random values in table 3.4 gives the values $y_1 = 2$ and $y_2 = 2$.



3.5

We start by transforming the first pseudorandom number into a value of Y_1 . The distribution function is given by

$$F_{Y_1}(x) = \int_{-\infty}^x f_{Y_1}(x) dx = \begin{cases} 0 & \text{if } x < 0, \\ 0.05x & \text{if } 0 \leq x \leq 5, \\ 0.15x - 0.5 & \text{if } 5 < x \leq 10, \\ 1 & \text{if } 10 \leq x; \end{cases}$$

hence, $y_1 = F_{Y_1}^{-1}(0.22) = 4.4$. Given this result, the distribution function of Y_2 is given by

$$F_{Y_2}(x | Y_1 \leq 5) = \int_{-\infty}^x f_{Y_2}(x | Y_1 \leq 5) dx = \begin{cases} 0 & \text{if } x < 0, \\ 0.08x & \text{if } 0 \leq x \leq 5, \\ 0.12x - 0.2 & \text{if } 5 < x \leq 10, \\ 1 & \text{if } 10 \leq x. \end{cases}$$

Now, we get $y_2 = F_{Y_2}^{-1}(0.52 | Y_1 \leq 5) = 6.0$.

3.6

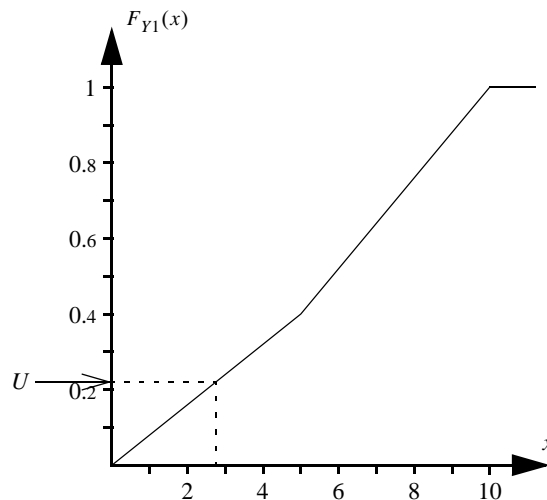
Let us start by randomising a value for the first input. To apply the inverse transform method, we need the distribution function $F_{Y_1}(x_1)$, which can be found by integrating the density function $f_{Y_1}(x_1)$. To find the density function of a single variable in a bivariate distribution, we need to compute

$$f_{Y_1}(x_1) = \int_{-\infty}^{\infty} f_Y(x_1, x_2) dx_2.$$

However, in this case the bivariate distribution is rather straightforward, and it is easy to see that all states of Y_1 in the interval $(0, 5)$ are equally probable and the same goes for all states in the interval $(5, 10)$. We can also see that the total probability that $Y_1 < 5$ is 40%, because

$$P(Y_1 < 5) = \int_{-\infty}^5 \int_{-\infty}^{\infty} f_Y(x_1, x_2) dx_2 dx_1 = 0.008 \cdot 5 \cdot 10 = 0.4.$$

This results in the distribution function shown below. Applying the inverse transform method on the random number 0.22 results in the value $y_1 = 2.75$.



Now, we need the probability distribution of Y_2 given that the outcome of the first variable is equal to 2.75, i.e. we need to compute

$$f_{Y_2|Y_1=2.75}(x_2) = \frac{f_Y(2.75, x_2)}{f_{Y_1}(2.75)}.$$

Fortunately, we may observe that $f_Y(2.75, x_2)$ has a constant value in the interval $(0, 10)$ and is equal to zero otherwise; hence, Y_2 must be uniformly distributed between 0 and 10. Applying the inverse transform method on the random number 0.64 then yields $y_1 = 6.4$.

4.1

$$\text{a) } m_X = \frac{1}{500} \sum_{i=1}^{500} x_i = 0.496.$$

$$\text{b) } s_{\bar{X}}^2 = \frac{1}{500} \sum_{i=1}^{500} (x_i - m_X)^2 = \frac{1}{500} \sum_{i=1}^{500} x_i^2 - m_X^2 = 0.258.$$

$$\text{c) } a_X = \frac{s_X}{m_X \sqrt{n}} = \frac{\sqrt{0.258}}{0.496 \sqrt{500}} \approx 0.0458.$$

d) $a_X > 0.01 \Rightarrow$ The stopping criteria is not fulfilled and the simulation should continue.

$$\text{e) } \delta = \frac{t_{\alpha} \cdot s_X}{\sqrt{n}} \approx \{t_{0.95} = 1.9600\} \approx 0.045 \Rightarrow 0.496 \pm 0.045 \text{ is a 95\% confidence interval for } E[X].$$

4.2

$$\text{a) } m_X = \frac{1}{100} \sum_{i=1}^{100} x_i = 200.$$

$$\text{b) } s_{\bar{X}}^2 = \frac{1}{100} \sum_{i=1}^{100} (x_i - m_X)^2 = \frac{1}{100} \sum_{i=1}^{100} x_i^2 - m_X^2 = 0.$$

c) The estimated variance is equal to zero, which means that all the 100 collected samples produced the same result. There are two possible explanations: either X is not random (and then there would not be any point in sampling X) or X is duogeneous and only conformist units have been sampled so far. In the latter case it is not appropriate to stop the simulation until some diverging units have been sampled. As the coefficient of variation will be equal to zero when the estimated variance is zero—because $a_X = s_X / (m_X \sqrt{n})$ —we need some additional criteria in the stopping rule, for example that $s_X > 0$.

4.3

$m_X \pm \delta$, where $\delta = \frac{t_{0.95} \cdot s_X}{\sqrt{n}}$, is a 95% confidence interval. If $\delta \leq 0.1m_X$ then we get

$$\frac{t_{0.95} \cdot s_X}{\sqrt{n}} \leq 0.1m_X;$$

hence

$$a_X = \frac{s_X}{m_X \sqrt{n}} \leq \frac{0.1}{t_{0.95}}.$$

Consequently, $\rho = 0.1/1.9600 \approx 0.05$ is an appropriate relative tolerance.

5.1

Symmetrical distribution \Rightarrow If $Y = \mu_Y + \delta$ then $Y^* = \mu_Y - \delta \Rightarrow y^* = 175$.

5.2

Symmetrical distribution \Rightarrow If $Y = \mu_Y + \delta$ then $Y^* = \mu_Y - \delta \Rightarrow y^* = 68$.

5.3

$$\text{a) } y = F_Y^{-1}(0.80) = 9, y^* = F_Y^{-1}(1 - 0.80) = 3.$$

$$\text{b) } y = F_Y^{-1}(0.14) = 2.4, y^* = F_Y^{-1}(1 - 0.14) = 9.3.$$

$$\text{c) } y = F_Y^{-1}(0.42) = 6.2, y^* = F_Y^{-1}(1 - 0.42) = 7.8.$$

5.4

The dagger cycle length is $S = \text{floor}(1/p) = \{p = 0.28\} = 3 \Rightarrow$

$$F_{Y_j}^\ddagger(x) = \begin{cases} 20 & \text{if } 0.28(j-1) \leq x < 0.28j, \\ 10 & \text{otherwise,} \end{cases} \quad j = 1, 2, 3.$$

$$\text{a) } y_j = F_{Y_j}^\ddagger(0.66) = \begin{cases} 10 & j = 1, \\ 10 & j = 2, \\ 20 & j = 3. \end{cases}$$

$$\text{b) } y_j = F_{Y_j}^\ddagger(0.85) = \begin{cases} 10 & j = 1, \\ 10 & j = 2, \\ 10 & j = 3. \end{cases}$$

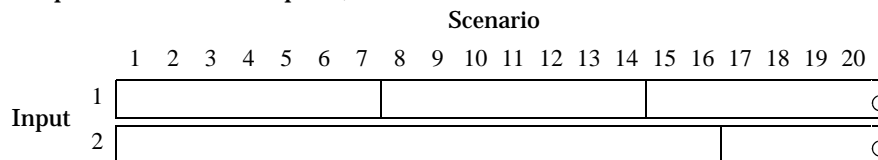
$$\text{c) } y_j = F_{Y_j}^\ddagger(0.04) = \begin{cases} 20 & j = 1, \\ 10 & j = 2, \\ 10 & j = 3. \end{cases}$$

5.5

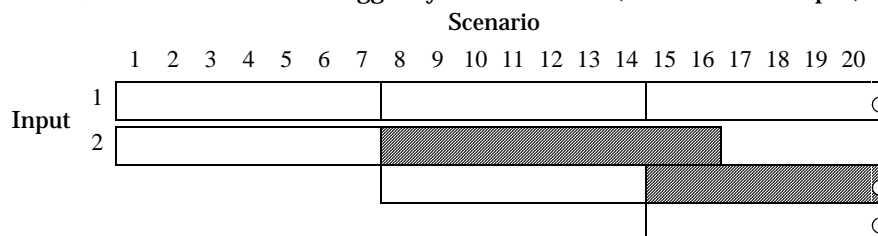
a) Let $u_i, i = 1, \dots, 20$ be twenty values from a pseudorandom number generator. Applying the inverse transform method to the joint probability distribution, $y_i = F_Y^{-1}(u_i)$ gives us 20 values of both inputs, i.e., 20 scenarios. Hence, 20 values from the random number generator are needed.

b) Let $u_i, i = 1, \dots, 10$ be ten values from a pseudorandom number generator, and let $u_i^* = 1 - u_i, i = 1, \dots, 10$. Applying the inverse transform method to the joint probability distribution, $y_i = F_Y^{-1}(u_i)$ and $y_i^* = F_Y^{-1}(u_i^*)$ gives us 20 values of both inputs, i.e., 20 scenarios. Hence, 10 values from the random number generator are needed.

c) Each dagger cycle will require one value from the random number generator. The dagger cycles are represented by rectangles in the figure below. As can be seen, there is a need for five dagger cycles in total (three for input 1 and two for input 2).



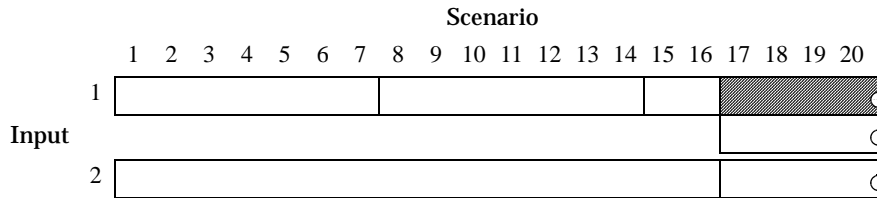
d) The shaded areas below are the part of the longer dagger cycle that is ignored when the dagger cycle is reset. In this case, the total number of dagger cycles will be six (three for each input).



e) The shaded areas below are the part of the shorter dagger cycle that is ignored when the dagger cycle is reset. In this case, the total number of dagger cycles will be six (four for input 1 and two for input 2).

5.6

a) In each trial there is an 85% probability that we get the value 1; hence, for six independent trials the probability that the result is equal to one every time is $0.85^6 \approx 37.7\%$.



b) Three pseudorandom numbers are needed, and each time there will be a 70% probability that we get the value 1 for both the original and complementary random number; hence, the probability that the result is equal to one every time is $0.7^3 = 34.3\%$.

c) The dagger cycle length will be equal to six; hence, we will be generating exactly one dagger cycle and the probability that we get only the result one is equal to the probability that the pseudorandom number falls within the rest interval, i.e., 10%.

5.7

a) The inverse distribution function is given by

$$F_Y^{-1} = \begin{cases} 0 & \text{if } 0 \leq x \leq 0.1, \\ 1 & \text{if } 0.1 < x \leq 1. \end{cases}$$

It is then easy to see that all the random numbers in table 5.4 will be transformed into the value 1, i.e., $y_i = 1, i = 1, \dots, 10$.

b) Apply the inverse transform using only five values from table 5.4, $u_i, i = 1, \dots, 5$, whereas the remaining five values to be transformed are given by $1 - u_i, i = 1, \dots, 5$. The original values will be equal to 0 if u_i is less than 0.1, whereas the complementary random numbers will be equal to 0 if u_i is larger than 0.9; however, there are no such entries, which means that we will get $y_i = 1, i = 1, \dots, 10$, in this case as well.

c) The dagger cycle length is given by $S = \text{floor}(1/p) = \{p = 0.1\} = 10$; thus, we will only need to generate one dagger cycle (which means that we are only going to need one random number). The first value in table 5.4 is 0.44, which points to the fifth interval, i.e., the condition $p(j-1) \leq x < p \cdot j$ in the dagger transform is fulfilled for $j = 5$. The resulting sequence is then that $y_i = 1, i = 1, \dots, 4, 6, \dots, 10$ and $y_5 = 0$.

d) With the inverse transform method and complementary random numbers, the value 1 does not appear in the generated sequence, whereas for dagger sampling it appears in one out of ten scenarios, which is exactly according to the actual probability distribution of Y . The result of dagger sampling is not a coincidence, as there is no rest interval in the dagger transform and we are generating a complete dagger cycle! For other pseudorandom numbers, the inverse transform method and complementary random numbers could have generated one or more results where $y_i = 0$. However, the probability of getting the results in questions a and a are not insignificant ($0.9^{10} \approx 35\%$ for the inverse transform method and $0.8^5 \approx 33\%$ for complementary random numbers).

5.8

In order to compute the estimate, we need to calculate the input density function $f_{Y_j}(y_j), j = 1, 2, 3$, the importance sampling function $f_{Z_j}(y_j), j = 1, 2, 3$, and the output x for the five scenarios:

| Scenario | $f_{Y_1}(y_1)$ | $f_{Z_1}(y_1)$ | $f_{Y_2}(y_2)$ | $f_{Z_2}(y_2)$ | $f_{Y_3}(y_3)$ | $f_{Z_3}(y_3)$ | $w = \frac{f_{Y_1}(y_1)f_{Y_2}(y_2)}{f_{Z_1}(y_1)f_{Z_2}(y_2)}$ | x |
|----------|----------------|----------------|----------------|----------------|----------------|----------------|---|-----|
| 1 | 0.9 | 0.6 | 0.9 | 0.6 | 0.9 | 0.6 | 3.375 | 0 |
| 2 | 0.1 | 0.4 | 0.9 | 0.6 | 0.9 | 0.6 | 0.5625 | 1 |
| 3 | 0.9 | 0.6 | 0.1 | 0.4 | 0.9 | 0.6 | 0.5625 | 0 |
| 4 | 0.9 | 0.6 | 0.9 | 0.6 | 0.9 | 0.6 | 3.375 | 0 |
| 5 | 0.9 | 0.6 | 0.9 | 0.6 | 0.1 | 0.4 | 0.5625 | 0 |

Then, $E[X]$ is estimated by

$$m_X = \frac{1}{5} \sum_{i=1}^5 w_i x_i = 0.5625/5 = 0.1125.$$

5.9

The weight factors for the five scenarios are given by

$$w_i = \frac{f_{Y1}(y_1, i) \cdot f_{Y2}(y_2, i)}{f_{Z1}(y_1, i) \cdot f_{Z2}(y_2, i)} = \begin{cases} 0.08 & i = 1, \\ 0.32 & i = 2, \\ 0.72 & i = 3, \\ 0.72 & i = 4, \\ 0.72 & i = 5. \end{cases}$$

The estimated expectation value is then given by

$$m_X = \frac{1}{5} \sum_{i=1}^n w_i x_i \approx 103$$

and the estimated variance is

$$s_X^2 = \frac{1}{5} \left(\sum_{i=1}^n w_i x_i^2 \right) - m_X^2 \approx 10\,808.$$

5.10

The inputs have nine possible states, and the expectation value $E[X] = \mu_X$ can be calculated by enumeration:

$$\mu_X = \sum_{i=1}^9 f_Y(\psi_i) g(\psi_i) = \dots = 20,$$

where f_Y is the joint probability distribution of both inputs. Then, the importance sampling function should be chosen as

$$f_Z(\psi) = \frac{g(\psi) f_Y(\psi)}{\mu_X};$$

the resulting optimal importance function is computed in the table below:

| Y_1 | Y_2 | f_Y | $X = g(Y_1, Y_2)$ | f_Z |
|-------|-------|-------|-------------------|--------|
| 0 | 0 | 0.05 | 0 | 0 |
| 0 | 5 | 0.15 | 5 | 0.0375 |
| 0 | 10 | 0.05 | 20 | 0.0500 |
| 5 | 0 | 0.10 | 5 | 0.0250 |
| 5 | 5 | 0.30 | 20 | 0.3000 |
| 5 | 10 | 0.10 | 30 | 0.1500 |
| 10 | 0 | 0.05 | 20 | 0.0500 |
| 10 | 5 | 0.15 | 30 | 0.2250 |
| 10 | 10 | 0.05 | 65 | 0.1625 |

5.11

Choose the importance sampling function

$$f_Z(\psi) = \frac{\tilde{g}(\psi)f_Y(\psi)}{\mu_Z},$$

where

$$\begin{aligned} \mu_Z = E[\tilde{g}(Y)] &= \int_{-\infty}^{\infty} f_Y(x)\tilde{g}(x)dx = \int_0^5 0.05xdx + \int_5^{10} 0.10xdx + \int_{10}^{15} 0.05(2x-10)dx \\ &= 0.625 + 3.750 + 3.750 = 8.125. \end{aligned}$$

Hence, we get

$$f_Z(\psi) = \begin{cases} 0 & \psi < 0, \\ \frac{0.05}{8.125}\psi & 0 \leq \psi \leq 5, \\ \frac{0.10}{8.125}\psi & 5 < \psi \leq 10, \\ \frac{0.05}{8.125}(2\psi - 10) & 10 < \psi \leq 15, \\ 0 & 15 < \psi. \end{cases}$$

5.12

The expectation values of the simplified models are

$$\mu_{Z1} = \sum_{\psi=1}^5 f_Y(\psi)z_1(\psi) = 400, \quad \mu_{Z2} = \sum_{\psi=1}^5 f_Y(\psi)z_2(\psi) = 0.1.$$

The optimal importance function for each output is then given by

$$f_{Z1}(\psi) = \frac{f_Y(\psi)z_1(\psi)}{\mu_{Z1}} = \begin{cases} 0.0250 & \psi = 1, \\ 0.1125 & \psi = 2, \\ 0.3500 & \psi = 3, \\ 0.2875 & \psi = 4, \\ 0.2250 & \psi = 5, \\ 0 & \text{all other } \psi, \end{cases}$$

$$f_{Z2}(\psi) = \frac{f_Y(\psi)z_2(\psi)}{\mu_{Z2}} = \begin{cases} 1 & \psi = 5, \\ 0 & \text{all other } \psi. \end{cases}$$

A reasonable compromise of these two importance sampling functions is to use the average, i.e.,

$$f_Z(\psi) = \frac{f_{Z1}(\psi) + f_{Z2}(\psi)}{2} = \begin{cases} 0.0125 & \psi = 1, \\ 0.0563 & \psi = 2, \\ 0.1750 & \psi = 3, \\ 0.1437 & \psi = 4, \\ 0.6125 & \psi = 5, \\ 0 & \text{all other } \psi. \end{cases}$$

5.13

The importance sampling function resulting in the lowest value of $\text{Var}[M_X]$ will be the most efficient, as the computation time is the same for both methods. The variance of the estimate is given by

$$\text{Var}[M_X] = \frac{1}{n} \left(\sum_{\psi \in \mathcal{Y}} g^2(\psi) \frac{f_Y^2(\psi)}{f_Z(\psi)} - \mu_X^2 \right),$$

where \mathcal{Y} is the set of possible input values. Enumeration of \mathcal{Y} yields the following results:

| ψ | | | $g(\psi)$ | $f_Y(\psi) = f_{Y1}(y_1)f_{Y2}(y_2)f_{Y3}(y_3)$ | $f_Z(\psi) = f_{Z1}(y_1)f_{Z2}(y_2)f_{Z3}(y_3)$ | $f_Z(\psi) = f_{Z2}(y_1)f_{Z2}(y_2)f_{Z2}(y_3)$ |
|--------|-------|-------|-----------|---|---|---|
| y_1 | y_2 | y_3 | | | | |
| 0 | 0 | 0 | 0 | 0.001 | 0.125 | 0.008 |
| 0 | 0 | 1 | 0 | 0.009 | 0.125 | 0.032 |
| 0 | 1 | 0 | 0 | 0.009 | 0.125 | 0.032 |
| 0 | 1 | 1 | 0 | 0.081 | 0.125 | 0.128 |
| 1 | 0 | 0 | 0 | 0.009 | 0.125 | 0.032 |
| 1 | 0 | 1 | 1 | 0.081 | 0.125 | 0.128 |
| 1 | 1 | 0 | 1 | 0.081 | 0.125 | 0.128 |
| 1 | 1 | 1 | 1 | 0.729 | 0.125 | 0.512 |

Hence, the variances are given by

$$\text{Var}[M_X] \approx \frac{4.36 - \mu_X^2}{n} \quad (\text{for the first method})$$

and

$$\text{Var}[M_X] \approx \frac{1.14 - \mu_X^2}{n} \quad (\text{for the second method}).$$

The true expectation value μ_X does not depend on the importance sampling function; therefore, we can conclude that the second method (i.e., using the importance sampling function f_{Z2}) is more efficient.

5.14

a) First we estimate the expectation value for each stratum:

$$m_{Xh} = \frac{1}{n_h} \sum_{i=1}^{n_h} x_{h,i} = \begin{cases} 200 & h = 1, \\ 500 & h = 2, \\ 550 & h = 3, \\ 1\,000 & h = 4. \end{cases}$$

The estimate for the entire population is obtained by

$$m_X = \sum_{h=1}^4 \omega_h m_{Xh} = 440.$$

b) The estimated variance in each stratum is given by

$$s_{Xh}^2 = \frac{1}{n_h} \sum_{i=1}^{n_h} (x_{h,i} - m_{Xh})^2 = \frac{1}{n_h} \sum_{i=1}^{n_h} x_{h,i}^2 - m_{Xh}^2 = \begin{cases} 1\,000\,000 & h = 1, \\ 2\,250\,000 & h = 2, \\ 3\,062\,500 & h = 3, \\ 250\,000 & h = 4. \end{cases}$$

According to the Neyman allocation, the target distribution of 2 000 samples should be

$$n_h^{\circledast} = n \frac{\omega_h^{S_{Xh}}}{\sum_{k=1} \omega_k^{S_{Xk}}} = \begin{cases} 640 & h = 1, \\ 720 & h = 2, \\ 560 & h = 3, \\ 80 & h = 4. \end{cases}$$

Considering the number of samples already collected from each stratum, 390 samples are needed from stratum 1, as well as 320 samples from stratum 2, 260 samples from stratum 3 and 30 samples from stratum 4.

5.15

According to the Neyman allocation, the target distribution of 200 samples is

$$n_h^{\circledast} = n \frac{\omega_h^{S_{Xh}}}{\sum_{k=1} \omega_k^{S_{Xk}}} \approx \begin{cases} 21 & h = 1, \\ 7 & h = 2, \\ 28 & h = 3, \\ 144 & h = 4. \end{cases}$$

The first two strata have apparently been allocated too many samples already in the pilot study. Hence, we should use a compromise allocation that allocates the next 100 samples to strata 3 and 4. We would like to have $119 + 3 = 122$ samples in strata 3 and 4 for the next batch. Consequently, $22/122 \approx 18\%$ of the samples cannot be collected. If this scarcity is shared by the two strata which need more samples, the resulting allocation will be 3 samples from stratum 3 and 97 samples from stratum 4. (If rounded values are used in these calculations—instead of the full precision of the calculator—the result can be 2 samples from stratum 3 and 98 from stratum 4.)

5.16

According to the Neyman allocation, the target distribution of 2 000 samples is

$$n_h^{\circledast} = n \frac{\omega_h^{S_{Xh}}}{\sum_{k=1} \omega_k^{S_{Xk}}} = \begin{cases} 700 & h = 1, \\ 900 & h = 2, \\ 400 & h = 3, \end{cases}$$

for the first output, and

$$n_h^{\circledast} = n \frac{\omega_h^{S_{Xh}}}{\sum_{k=1} \omega_k^{S_{Xk}}} = \begin{cases} 0 & h = 1, \\ 900 & h = 2, \\ 1\,100 & h = 3, \end{cases}$$

for the second. Using the mean of these two distributions as a compromise allocation and taking into account the scenarios that already have been generated, we should choose to generate 150 scenarios in stratum 1, 500 scenarios in stratum 2, and 350 scenarios in stratum 3.

5.17

a) $\omega_1 = P(Y_1 \leq 3) \cdot P(Y_2 \geq 3) = F_{Y_1}(3) \cdot (1 - F_{Y_2}(3)) = 0.12$. Similarly, $\omega_2 = (1 - F_{Y_1}(3)) \cdot (1 - F_{Y_2}(3)) = 0.28$, $\omega_3 = F_{Y_1}(3) \cdot F_{Y_2}(3) = 0.18$ and $\omega_4 = (1 - F_{Y_1}(3)) \cdot F_{Y_2}(3) = 0.42$.

b) We start by sorting out which scenario belongs to which stratum and then we calculate the estimated

expectation value for each stratum. The result is as follows:

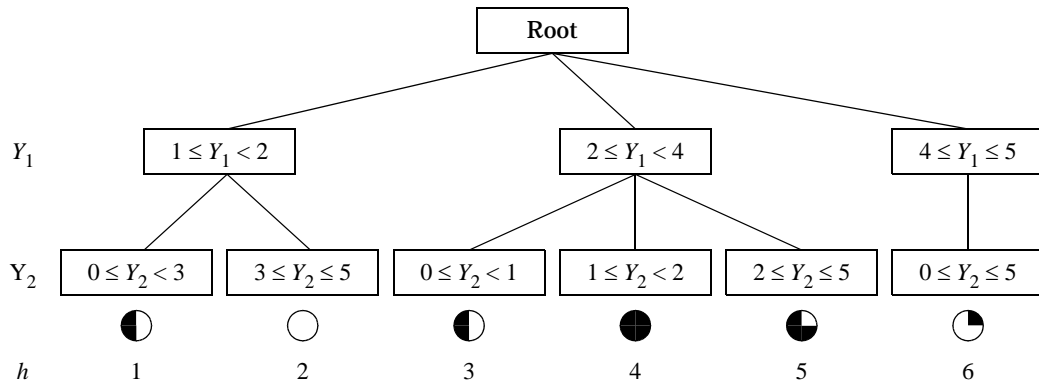
| Stratum, h | Scenarios, i | Estimated expectation value, $m_{Xh} = \frac{1}{n_h} \sum_{i=1}^{n_h} x_{h,i}$ |
|--------------|----------------|---|
| 1 | 2, 4 | 7.1 |
| 2 | 5, 6, 8, 10 | 15.2 |
| 3 | 1, 3 | 2.7 |
| 4 | 7, 9 | 6.5 |

The estimate for the entire population is obtained by

$$m_X = \sum_{h=1}^4 \omega_h m_{Xh} = 8.324.$$

5.18

a) To determine the shape of the area we can investigate the share of points within the rectangle $1 \leq Y_1 \leq 5$ and $0 \leq Y_2 \leq 5$. The idea is to separate areas which are either clearly inside or clearly outside the shape from areas which are both inside and outside the shape. One possible strata tree according to this principle is shown below. The circles below each branch indicate the expected properties of the corresponding stratum.



b) Both inputs should be uniformly distributed; hence, the stratum weight will be equal to the size of stratum divided by the size of the rectangle which comprises the sample space. Thus, we get the following stratum weights:

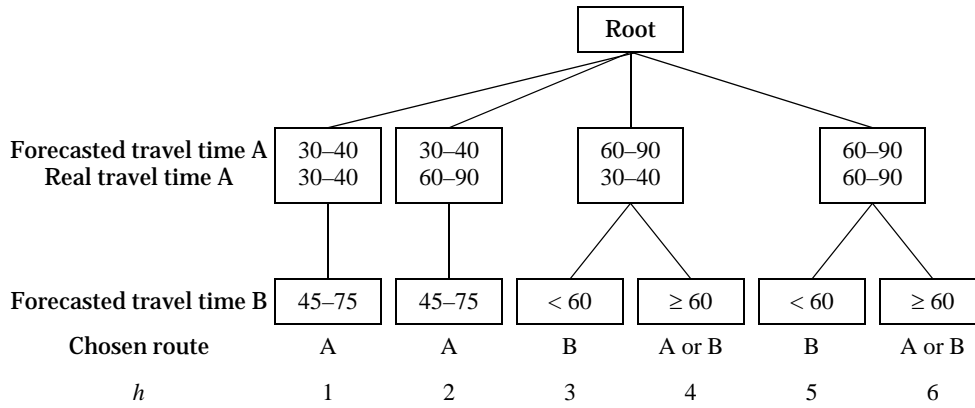
$$\omega_1 = 1 \cdot 3/20 = 0.15, \quad \omega_2 = 1 \cdot 2/20 = 0.1, \quad \omega_3 = 2 \cdot 1/20 = 0.1,$$

$$\omega_4 = 2 \cdot 1/20 = 0.1, \quad \omega_5 = 2 \cdot 3/20 = 0.3, \quad \omega_6 = 1 \cdot 5/20 = 0.25.$$

5.19

a) We can use a strata tree to identify scenarios where Alice's choice of route is easily predicted. We put the situation (which in practice is an auxiliary input) on the first level below the root and the forecasted travel time of route B on the second level, as in the figure below. In the first stratum, the forecasted travel time of route A is always smaller than the forecasted travel time of route B; hence, Alice will choose route A. This is also the case in the second stratum, but the difference is that in this case the forecast is highly inaccurate and the resulting travel time will be much longer compared to the scenarios in the first stratum. In the third and fifth strata, the forecasted travel time of route B is always smaller than the forecast for route A, and Alice will go route B. In the remaining strata, Alice's choice cannot be

predicted.¹



b) The expectation values of strata 1, 2, 3 and 5 can be computed analytically. For example, in stratum 1 we know that Alice will choose route A and that the expected real travel time is 35 minutes. Similarly, in stratum 3 Alice will go route B and the expected real travel time will then be 60 minutes (the real travel time on route B is independent of the forecast and will consequently be $U(45, 75)$ -distributed regardless of whether or not the forecasted travel time is less than 60 minutes).

This means that there are only two strata, where the expectation value need to be estimated. A reasonable solution is to allocate the same amount of scenarios in the pilot study to each stratum, i.e., 50 scenarios should be generated in stratum 4 and another 50 scenarios should be generated in stratum 6.

5.20

a) Before we can investigate the variances of the two estimates, we need the properties of the variables to be sampled:

$$\mu_X = \sum_{x_1} \sum_{x_2} f_{Y1}(x_1) f_{Y2}(x_2) g(x_1, x_2) = 2.87,$$

$$\sigma_X^2 = \sum_{x_1} \sum_{x_2} f_{Y1}(x_1) f_{Y2}(x_2) (g(x_1, x_2) - \mu_X)^2 = 5.13,$$

$$\mu_{X1} = \sum_{x_1} f_{Y1}(x_1) g(x_1, 0) = 10, \quad \sigma_{X1}^2 = \sum_{x_1} f_{Y1}(x_1) (g(x_1, 0) - \mu_{X1})^2 = 10,$$

$$\mu_{X2} = \sum_{x_1} f_{Y1}(x_1) g(x_1, 2) = 2.8, \quad \sigma_{X2}^2 = \sum_{x_1} f_{Y1}(x_1) (g(x_1, 2) - \mu_{X2})^2 = 4.56.$$

We also need the stratum weights:

$$\omega_1 = P(Y_2 = 0) = f_{Y2}(0) = 0.01, \quad \omega_2 = P(Y_2 = 2) = f_{Y2}(2) = 0.99.$$

The variance of the estimate when using simple sampling would be

$$\text{Var}[M_X] = \frac{\sigma_X^2}{n} = \frac{5.13}{n}.$$

This should be compared to the variance of the estimate when using stratified sampling, which is given by

$$\text{Var}[M_X] = \sum_{h=1}^2 \omega_h^2 \frac{\sigma_{Xh}^2}{n/2} = \frac{2 \cdot 0.01^2 \cdot 10 + 2 \cdot 0.99^2 \cdot 4.56}{n} = \frac{8.94}{n}.$$

1. It could be noted that strata 4 and 6 could be further divided to identify scenarios where the forecasted travel time on route A is larger than 75, in which case we know that Alice will choose route B. However, this strata tree is sufficient for this problem.

Assuming that the differences in computation time are negligible, the variance of the estimate is lower for simple sampling; hence, this application of stratified sampling was not efficient.

b) The efficiency of stratified sampling can be improved by changing the strata definition (so that strata are more homogenous) or by applying the Neyman allocation instead of proportional sampling.

5.21

We start by calculating the estimated difference between the detailed model and the control variate for each stratum:

$$m_{(X-Z)h} = \frac{1}{200} \sum_{i=1}^h ((x_{h,i} + x_{h,i}^*) - (z_{h,i} + z_{h,i}^*)) = \begin{cases} 1.25 & h = 1, \\ 4.5 & h = 2, \\ 8.25 & h = 3. \end{cases}$$

Then we calculate the estimated difference between the detailed model and the control variate for the entire population:

$$m_{(X-Z)} = \sum_{h=1}^3 \omega_h m_{(X-Z)h} = 3.625.$$

Finally, we add the expectation value of the control variate to get the estimated expectation value of the detailed model:

$$m_X = m_{(X-Z)} + \mu_Z = 12.625.$$

5.22

The variable to be studied is the difference $X_1 - X_2$. However, since importance sampling is used, each sample should be multiplied by the weight factor. This produces two estimates

$$m_{(X_1 - X_2), 1} = \frac{1}{500} \sum_{i=1}^{500} w_i (x_{1,i} - x_{2,i})$$

and

$$m_{(X_1 - X_2), 2} = \frac{1}{500} \sum_{i=1}^{500} w_i^* (x_{1,i}^* - x_{2,i}^*).$$

The final estimate is the mean of these two, i.e.,

$$\begin{aligned} m_{(X_1 - X_2)} &= \frac{1}{1\,000} \sum_{i=1}^{500} (w_i (x_{1,i} - x_{2,i}) + w_i^* (x_{1,i}^* - x_{2,i}^*)) = \\ &= \frac{1}{1\,000} \left(\sum_{i=1}^{500} w_i x_{1,i} + \sum_{i=1}^{500} w_i^* x_{1,i}^* - \sum_{i=1}^{500} w_i x_{2,i} - \sum_{i=1}^{500} w_i^* x_{2,i}^* \right) = \\ &= (50\,300 + 49\,600 - 49\,100 - 48\,800) / 1\,000 = 2\,000 / 1\,000 = 2. \end{aligned}$$

5.23

We start by calculating the estimated expectation value for each stratum:

$$m_{Xh} = \frac{1}{n_h} \sum_{i=1}^{n_h/2} \left(\frac{f_Y(y_{h,i})}{f_Z(y_{h,i})} g(y_{h,i}) + \frac{f_Y(y_{h,i}^*)}{f_Z(y_{h,i}^*)} g(y_{h,i}^*) \right) = \begin{cases} 16 & h = 1, \\ 21 & h = 2, \\ 34 & h = 3. \end{cases}$$

Then we calculate the estimate for the entire population:

$$m_X = \sum_{h=1}^3 \omega_h m_{Xh} = 21.1.$$

5.24

The results for the ten scenarios are given in the table below.

| Scenario | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------|---|---|---|---|---|---|---|---|---|----|
| $g_1(z)$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| $g_2(z)$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

The expected difference between the two systems is estimated by

$$m_{(X_2 - X_1)} = \frac{1}{n} \sum_{i=1}^n w(z_i) (g_2(z_i) - g_1(z_i)).$$

In this case, the difference $g_2(z_i) - g_1(z_i)$ is equal to zero for all scenarios except the eighth, i.e.,

$$m_{(X_2 - X_1)} = \frac{1}{10} w(z_8) = \frac{1}{10} \left(\frac{0.9}{0.9} \cdot \frac{0.9}{0.9} \cdot \frac{0.9}{0.5} \cdot \frac{0.1}{0.5} \cdot \frac{0.1}{0.5} \cdot \frac{0.9}{0.9} \right) = 0.007.$$