Monte Carlo Methods 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 thoery 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
September 2013

Mikael Amelin

CONTENTS

Chapter 1
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 expactation 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 Monaco[1]—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 figur 1. The studied systems are modelled by a set of ran-

[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”.

To be added: Figure...

Figure 1 Overview of the simulation problem studied in this compendium.

dom 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 $X = g(Y)$. The function g does not have to be expressed explicitly, can for example be derived from the solution to one—or even several—optimisation problems, as illustrated in the second of the following two examples:

> ***Example 1.1.*** To be added: Example of a model expressed explicitly...
>
> ***Example 1.2.*** To be added: Example of a model derived from the solution to an optimisation problem (multi-area economic dispatch problem)...

It is important to notice that the model g is deterministic! Hence, if we have two sets of input values, 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 sucess of reclosing breakers after a failure) ...

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 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 need 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 $\mathscr{y}_i$ or $\mathscr{x}_i$.
- **Probability distributions.** Probability distributions are denoted by Latin f in upper or lower case (depending on interpretation[2]) 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.

[2] Cf. section 2.1.

- **Estimates.** Estimates of key statistical properties for a probability distribution are denoted by upper or lower case (depending on interpretation[3]) 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.

[3] Cf. section 4.1.

Chapter 2

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 multidimensional. 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

KOMMENTAR: *Definiera täthetsfunktion, fördelningsfunktion och populationer (inklusive begreppen homogena, heterogena och duogena populationer).*

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 2.7. 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:

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

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 2.2 (The S:t 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 2.8.*** The variance of a random variable X is given by
>
> $$Var[X] = E[(X - E[X])^2] = E[X^2] - (E[X])^2 \text{ (general definition)},$$
>
> $$Var[X] = \frac{1}{N}\sum_{i=1}^{N}(x_i - E[X])^2 \text{ (population)},$$
>
> $$Var[X] = \sum_{x \in \Omega_X} f_X(x)(x - E[X])^2 \text{ (discrete random variable)},$$
>
> $$Var[X] = \int_{x \in \Omega_X} f_X(x)(x - E[X])^2 dx \text{ (continuous random variable)}.$$

A disadvantage with variance is that the unit of $Var[X]$ is the square of the unit of X. For example, if X is the unserved load expressed in MWh/h then the variance of X is expressed in (MWh/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 2.9.*** The standard deviation of a random variable X is given by
>
> $$\sigma_X = \sqrt{Var[X]}.$$

Covariance and Correlation Coefficient

To be added...

2.3 Arithmetics of Random Variables

KOMMENTAR: *Definiera grundläggande räkneregler för stokastiska variabler och deras statistiska egenskaper.*

Exercises

To be added...

Chapter 3

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:

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

$$U_i = \frac{X_i}{m}, \tag{3.1b}$$

where

X_1 = 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 of when dividing ($aX_i + c$) by m. The result of this operation is an integer in the interval 0, …, 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 divisable 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 are 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

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 particularly efficient for Monte Carlo simulations are described.

3.2.1 Independent Random Numbers

To be added: Introduction...

The Inverse Transform Method

To be added: Introduction...

> ***Theorem 3.1.*** If a random variable U is $U(0, 1)$-distributed then the random variable $Y = F_Y^{-1}(U)$ has the distribution function $F_Y(x)$.

To be added: Comments and examples...

Random Numbers from Finite Populations

To be added: Introduction...

To be added: Search algortihm...

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...

> ***Theorem 3.2.*** 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

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

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

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

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

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

and finally

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

To be added: Comments and exampels...

3.2.2 Correlated Random Numbers

A Monte Carlo simulation can have inputs that are correlated. It should however be noted that generation of correlated random numbers is not as straightforward as generation of independent random numbers.

General Method

To be added...

Random Numbers from Finite Populations

To be added...

Normally Distributed Random Numbers

To be added...

Approximative Method

To be added...

Exercises

To be added...

Chapter 4

SIMPLE SAMPLING

To be added: Introduction...

4.1 Principle

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

Given that an estimate is a random variable, we may study the probability distribution of the estimate and discuss what properties a good estimate should have.

To be added: Figure showing the difference between low and high accuracy as well as low and high precision.

To be added: Systematical errors, definitions of accuracy, $E[M_X] = E[X]$.

To be added: Random errors, definition of precision, $Var[M_X]$.

Replacement

4.2 Application

To be added: Introduction...

4.2.1 Estimating the Properties of Random Variables

Estimating Expectation Values and Variance

To be added: Introduction...

Theorem 4.1. (Law of Large Numbers): If $x_1, \ldots, x_n$ are independent observations of the random variable X then $E[X]$ can be estimated by

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

Proof: Let t_i denote the number of times that unit i appears in the samples; hence, 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 Bernouille-distributed, i.e, t_i is $B(n,\ 1/N)$-distributed.

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

$$n{\cdot}p\} = \sum_{i=1}^{N} \frac{1}{n}\frac{n}{N}\mathcal{X}_i = E[X]. \ ▮$$

Notice the similarity between the definition of expectation value (definition 2.7) and the formula for the estimate of the expectation value (theorem 4.1 above):

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

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. We will see that the same principle is applied when estimating other statistical properties.

Theorem 4.2. If $x_1, \ldots, x_n$ are independent observations of the random variable X then $Var[X]$ can be estimated by

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

Proof: To be added?

KOMMENTAR: *Visa hur beräkningen av* S_X^2 *kan förenklas till två summor…*

Estimating Probability Distributions

KOMMENTAR: *L3, exempel 8–9*

4.2.2 Precision of Estimates

To be added: Introduction...

Theorem 4.3. In simple sampling, the variance of the estimated expectation value is

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

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

$$Var[M_X] = \frac{Var[X]}{n}. \tag{4.1}$$

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

To be added: Confidence intervals...

Chapter 5

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 is 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

$$E[M_{X1}] = E[M_{X2}] = \mu_X. \tag{5.1}$$

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

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

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 (4.1) we have that simple sampling with in total n samples results in

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

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

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

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

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

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 questions 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.

If we want to apply complementary random numbers to the input Y in a Monte Carlo simulation of the system $X = g(Y)$ we will proceed as follows:

- 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 a 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

5.2 Dagger Sampling

5.3 Control Variates

5.4 Correlated Sampling

5.5 Importance Sampling

5.6 Stratified Sampling

5.7 Combining Several Variance Reduction Techniques

Chapter 6

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 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.

6.2 Planning a Monte Carlo Simulation

To be added: Suggested steps when setting up a Monte Carlo simulation

Appendix A

PROBABILITY DISTRIBUTIONS

This appendix provides an overview of some important probability distributions.

To be added: Reference to other sources.