Simulation handbook

W.M.P. van der Aalst and M. Voorhoeve
Dept. of Mathematics and Computer Science,
Technical University Eindhoven,
Postbox 513, 5600 MB, Eindhoven.

Recent developments in the area of hard- and software create new possibilities regarding simulation. High resolution monitors and pointing devices (mice) allow us to build simulation models quickly and clearly. Fast processors allow detailed simulation and even animation of long runs. However, these developments risk obliterating the need to analyze the reliability of the results generated. People used to look carefully at data from a simulation, whereas nowadays a smooth presentation eases away any second thoughts. As simulation studies often support important strategic decisions, there is a marked danger is such a development. It is unacceptable such decisions are based on unfounded results, so sufficient attention must be paid to the reliability of simulation. These thoughts led to the present handbook.

The Dutch version of this handbook by the first author has been used several years for teaching students from logistics and management at Eindhoven University of Technology. The growing international contacts at our university required an English version. This version was edited by the second author, who wishes to thank the Stan Ackermans Institute for financial support and Karin Luit for the translation.

Contents

Target group:
This handbook is intended for persons with little experience in statistics, yet involved in simulation studies. It can also be used for reference and as an addition to the manual for any simulation tools used.

1 Introduction

Suppose you own a large discotheque and are facing problems in deploying your staff on Saturday nights. On the one hand at certain times there is too much staff capacity, on the other hand customers complain about long waiting times for getting their coats hung up and ordering drinks. Because you feel you are employing too much staff and yet are facing the threat of losing customers due to excessive waiting times, you decide to make a thorough investigation. Examples of questions you want answered are:

- What are the average waiting times of customers at the different bars and at the cloakroom?
- What is the occupation rate of the bar staff?
- Will waiting times be reduced substantially if extra staff is deployed?
- Would it serve a purpose to deploy staff flexibly? (e.g. no longer assigning staff members to one bar only)
- What is the effect of introducing refreshment coupons on average waiting times?
- Will it serve a purpose to use waiters?
- What are the effects of introducing a 'happy hour' to spread the arrivals of guests?

Simulation model

To answer these and similar questions, a *simulation model* can be used. A simulation model reflects reality and thus can be used to simulate that reality in a computer. In the same way that an architect uses construction drawings to try and gain insight in the house modeled, a systems analyst may use simulation models to gain insight in the business process modeled.

When does it serve a purpose to set up a simulation model? Some obvious reasons are:

Reasons for simulation

- Gaining *insight* in an existing or proposed future situation. By charting a business process, it becomes apparent what is important and what is not.
- A real experiment is *too expensive*. Simulation is a cost-effective way to analyze several alternatives. Hiring extra staff or introducing a refreshment coupons system is too expensive to try out in reality if it does not lead to improvement. You want to know in

advance whether a certain measure will have the desired effect. Especially when starting up a new business process, simulation can save a lot of money.

- A real experiment is *too dangerous*. Some experiments cannot be carried out in reality. Before a railway company installs a new traffic guidance system, it must assess the safety consequences. The same holds for other processes where safety is critical (e.g. aviation or nuclear reactors).

Sometimes, rather than using simulation, a mathematical model, also called an analytical model is sufficient. In Operations Research (OR) many models have been developed which can be analyzed without simulation, such as queuing models, combined optimalization models, stochastic models, etc. Although the scope of these models is limited compared to simulation, why simulate if a simple analytical model can also do the job? In comparison to a simulation model, an analytical model has less detail and so requires less parameter data. Moreover, it requires much less resources (both time and computing power) to achieve the derivation of reliable conclusions.

Strong points of simulation models versus analytical models:

Strong points of simulation

- Simulation is flexible. Any situation, no matter how complex, can be investigated through simulation.
- Simulation can be used to answer widely divergent questions. It is possible to assess e.g. waiting times, occupation rates and fault percentages from one and the same model.
- Simulation is easy to understand. In essence, it is nothing but replaying a modeled situation. In contrast to many analytical models, little specialist knowledge is necessary to understand the model.

Unfortunately, simulation also has some disadvantages.

Weak points of simulation

- A simulation study can be very time consuming. Sometimes, very long simulation runs are necessary to achieve reliable results.
- One has to be very careful in interpreting simulation results. Determining the reliability of results can be very treacherous indeed.
- Simulation does not offer proofs. Whatever occurs in a correct simulation model may occur in reality, but the reverse does not hold. Things can happen in reality that are not witnessed during the simulation experiments.

Simulation is often used to numerically support strategic decisions. Simulation models can be rapidly constructed by using user-friendly simulation tools. However, a faulty model or a wrong interpretation of the results may lead to decisions without justification. Therefore, this handbook will focus on the validation of models made and the correct derivation and interpretation of simulation results.

This handbook is arranged as follows. Chapter 2 treats the construction of a simulation model. Chapter 3 will focus on the 'input side' of simulation, i.e. obtaining the necessary samples. A major part of this chapter will treat the various probability distributions that samples are drawn from. Chapter 4 focuses on the 'output side' of a simulation study: the interpretation of the results. Finally, chapter 5 reviews a number of frequent mistakes.

2 Constructing a simulation model

Simulation tools

For the construction of a simulation model we use a *tool*, which ensures that the computer can simulate the situation charted by the model. We can differentiate between two kinds of simulation tools:

Simulation languages A *simulation language* is a programming language with special provisions for simulation. Examples of simulation languages are SIMULA, GPSS, SIMSCRIPT, SIMPAS, SIMON, MUST and GASP.

Simulation packages A *simulation package* is a tool with building blocks for a certain application area, which allow the rapid creation of a simulation model, mostly graphically. Examples of simulation packages for production processes are: SIM-FACTORY, WITNESS and TAYLOR.

The advantage of a simulation language is that almost every situation can be modeled. The disadvantage is that one is forced to chart the situation in terms of a programming language. The modeling thus becomes time-consuming and the model itself provides little insight. A simulation package allows to rapidly build a clear model. Because the model must be built from ready-made building blocks, the area of application is limited. As soon as one transgresses the limits of the specific area of application, e.g. by changing the control structure, the modeling becomes cumbersome or even impossible.
In the past few years, tools have been introduced with characteristics of both a simulation language and a simulation package. Examples are ExSpect, Design/CPN, Extend and ARENA (formerly SIMAN/CINEMA).

These tools combine a graphic development environment to a programming language, also offering the possibility of animation. ExSpect and Design/CPN are based on high level Petri nets, allowing the use of analysis techniques apart from simulation. Tools such as Extend and ARENA are based on proprietary concepts, which makes them less suitable for further analysis. The use of proprietary building blocks also makes it hard to interchange simulation models between packages. Standardization based on a formal description method such as high-level Petri nets improves interchangeability between tools.

Phasing

The lack of a generally accepted method of description makes it impossible to give a generally applicable method of modeling. However, the *phasing* of a simulation study is more or less the same for all tools. Figure 1 shows the phases of a simulation study.

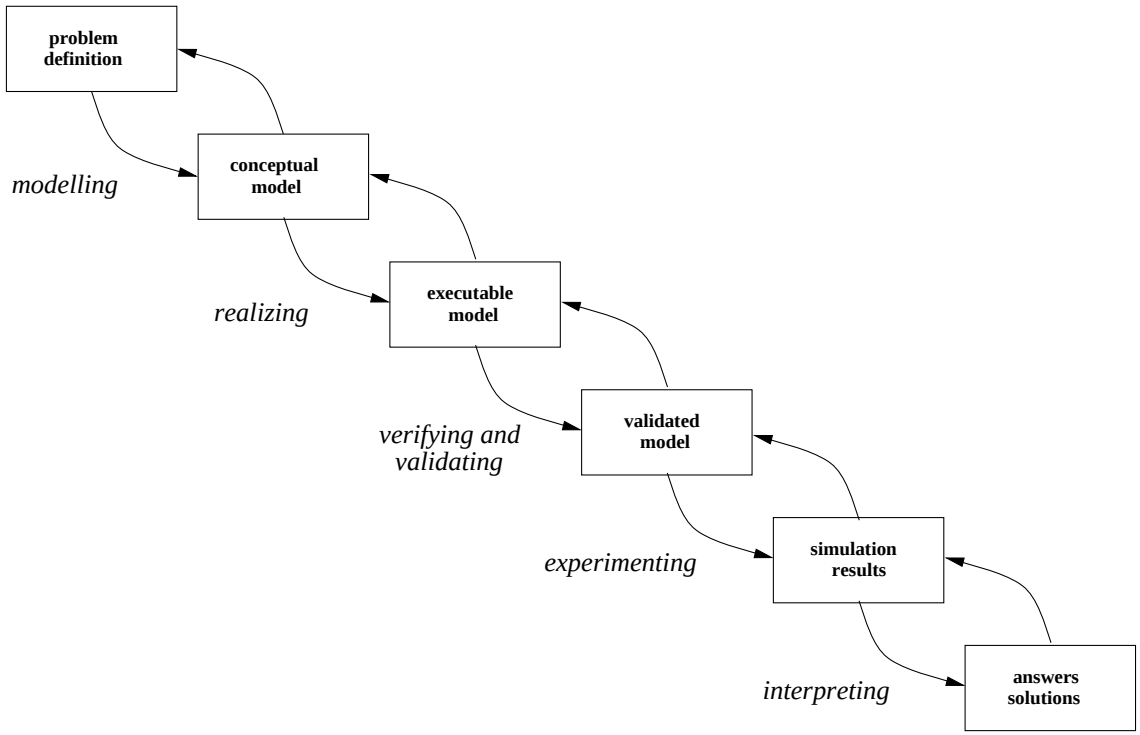

Figure 1: Phases of a simulation study.

Problem definition

The simulation process starts with a *problem definition* , describing the goals and fixing the scope of the simulation study. The scope tells what will and what will not be a part of the simulation model. The problem definition should also state the questions to be answered. Preferably, these questions should be quantifiable. Instead of asking "Are the customers satisfied?", one should ask "How long do customers have to wait on average?"

Modeling

Conceptual model

After defining the problem, the next phase is *modeling*. In this phase the *conceptual model* is created. The conceptual model defines classes of *objects* and the *relations* between these objects. In the case of the discotheque the objects to be distinguished are a/o. staff members, bars, cloakroom and customers. The relevant characteristics (properties) of these objects are charted. We distinguish between *qualitative* and *quantitative* characteristics. Examples of qualitative characteristics are queue disciplines (How are customers attended to, e.g. in order of arrival (FIFO - First-In-First-Out) or randomly? (SIRO - Select-In-Random-Order)), sequences (What actions are performed in what order when a customer enters?) and decision rules (When do we open an extra bar?). Quantitative characteristics of objects describe primarily speed (How many customers can be served per minute?) and capacity (How many customers can be served simultaneously?). If we are dealing with objects of the same class, we specify these characteristics for the entire class (parameterized if necessary). Graphs can be drawn for showing connections between the various objects or object classes. Suitable techniques are situation diagrams, data flow diagrams or simulation-specific flow diagrams.
The construction of the conceptual model will most likely unveil incomplete and contradictory aspects in the problem definition. Also, the modeling process may bring forth new questions for the simulation study to answer. In either case, the problem definition should be adjusted.

Realization

Executable model

After the conceptual modeling phase, the *realization phase* starts. Here, the conceptual model is mapped onto an *executable model*. The executable model can be directly simulated on the computer. How to create this model depends strongly on the simulation tool used. Simulation languages require a genuine design and implementation phase. Simulation packages that fit the problem domain merely require a correct parametrization. The objects of the conceptual model are mapped to building blocks from the package and their quantitative characteristics (e.g. speed) are translated to parameter values of these building blocks.

Verification

An executable model is not necessarily correct, so it has to be *verified*. Verification of the model is necessary to examine whether the model contains qualitative or quantitative errors, like programming errors or wrong parameter settings. For verification purposes, small trial runs can be simulated by hand and its results assessed, or a stress test can be applied to the model. In the stress test the model is subjected to extreme situations, like having more customers arrive than can be attended to. In such a case, waiting times measured should increase dramatically in the course of time. Some tools support more advanced forms of verification.

Using a Petri net based simulation tool, e.g. all kinds of logical properties of the model can be proven. For example, certain invariants (the number of staff members is constant) and other logical properties (absence of deadlock) can be inferred.

Validation

Apart from verification, *validation* of the model is also required. During validation we compare the simulation model with reality. When simulating an existing situation, the results of a simulation run can be compared to observations from historical data. Simulation models that relate to possible future situations, can be validated by comparing the simulation results to calculated results.

Verification and validation may lead to adjustments in the simulation model. New insights may even lead to adjusting the problem definition and/or the conceptual model. A simulation model found to be correct after validation is called a *validated model*.

Validated model

Experimenting

Starting from the validated model, *experiments* can be carried out. These experiments have to be conducted in such a way that reliable results are obtained as efficiently as possible. In this stage decisions will be made concerning the number of simulation runs and the length of each run.

Interpreting

The simulation results will have to be *interpreted*, to allow feedback to the problem definition. Reliability intervals will have to be calculated for the various measures gathered during simulation. Also, the results will have to be interpreted to answer the questions in the problem definition. For each such answer the reliability should be stated. All these matters are summarized in a final report with answers to questions from the problem definition and proposals for solutions.

Answers and solutions

Figure 1 shows that feedback is possible between phases. In practice, many phases do overlap. Specifically, experimentation and interpretation will often go hand in hand.

Alternatives

Figure 1 assumes the existence of a single simulation model. Usually, several *alternative situations* are compared to one another. In that case, several simulation models are created and experimented with and the results compared. Often, several possible improvements of an existing situation have to be compared through simulation. We call this a *what-if analysis*. In such a case a model of the current situation is made first. For this model the phases of Figure 1 are followed. The model is then repeatedly adjusted to represent the possible future situations. These adjustments may just concern the executable model (e.g. by changing parameters). In some cases (e.g. when changing control structures), the conceptual model is affected too. In each case, the adjustments should be validated. The different alternatives are experimented with and the

What-if analysis

results compared to indicate the expected consequences of each alternative.

User

Systems analyst

Programmer

The people involved in a simulation study have their specific responsibilities. In the first place, there are *users*: the persons confronted with the problem to be investigated. Secondly, there is a *systems analyst* , responsible for writing a clear problem definition. The analyst also creates the conceptual model. Depending on the tools used, the systems analyst can be supported by a *programmer* to realize the simulation model. The number of simulation experiments often dictates who should conduct them. If the experiments have to be conducted regularly, e.g. for supporting tactical decisions, a user seems appropriate. If it concerns a simulation study supporting a one-time strategic decision, the systems analyst or programmer is preferred. For the correct interpretation of the simulation results, it is of the utmost importance that the persons involved have sufficient knowledge of the statistical aspects of simulation.

Animation

The users and builders of a simulation model (the systems analyst and the programmer) are in general different people. Agreement between them is of prime importance, though. The eventual model should fit the ideas that the user had in mind. One way of promoting user involvement is by *animation*. Animation is the *graphical* simulation of the modeled situation from a simulation model, e.g. by moving objects that change shape. In this way an animated movie can be made of the model, that suits the user's experience. Although animation is a useful tool for e.g. the validation of a model by the user, one must take care. Pretty animations may cause the user to accept a model without questioning the quantitative results.

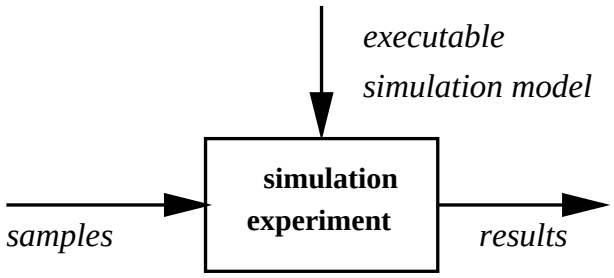

Figure 2: Input and output of a simulation experiment.

The remainder of this handbook will focus primarily on the statistical aspects of simulation experiments. Figure 2 shows the input and output of a simulation experiment. The input consists of an executable simulation model and random samples from various parameterized probability distributions. The output consists of (mostly numerical) simulation results.

Chapter 3 focuses on the random sampling necessary for the experiment. Chapter 4 deals with the interpretation of simulation results.

3 Sampling from distributions

3.1 Random and pseudo-random numbers

Environment

Monte Carlo method

A simulation experiment is little more than replaying a modeled situation. To replay this situation in computer, we have to make assumptions not only for the modeled system itself but also for its *environment*. As we cannot or will not model these matters in detail we turn to *Monte Carlo*! We do not know when and how many customers will enter a post office, but we do know the mean and variation of customer arrivals. So we have the computer take seemingly random samples from a probability distribution. The computer is by nature a deterministic machine, so techniques have been developed to generate so-called *pseudo-random numbers*.

Pseudo-random numbers

Random generator

A *random generator* is a piece of software for producing pseudo-random numbers. The computer does in fact use a deterministic algorithm to generate them, which is why they are called "pseudo" random. Most random generators generate pseudo-random numbers between 0 and 1. Each value between 0 and 1 being equally probable, these values are said to be distributed *uniformly* over the interval between 0 and 1. It depends on the random generator whether 0 and/or 1 themselves can be generated. This matter is of technical relevance only.
Most random generators generate a series of pseudo-random numbers $\frac{X_i}{m}$ according to the formula:

$$X_n = (aX_{n-1} + b) \text{ modulo } m$$

For each i, X_i is is a number from the set $\{0, 1, 2, \ldots, m-1\}$ and $\frac{X_i}{m}$ matches a sample from a uniform distribution between 0 and 1. The numbers a, b and m are chosen in such a way that the sequence can hardly or not at all be distinguished from 'truly random numbers'. This means that the sequence X_i must visit each of the numbers $0, 1, 2, \ldots, m-1$ once. Also, m is chosen as near as possible to the largest integer that can be manipulated directly by the computer hardware. There are several tests to check the quality of a random generator (cf. [2, 12, 14, 9]): frequency test, correlation test, run test, gap test and poker test.
A good random generator for a 32 bit computer is:

$$X_n = 16807X_{n-1} \text{ modulo } (2^{31} - 1)$$

That is: $a = 16807$, $b = 0$ and $m = 2^{31} - 1$. For a 36-bit machine:

$$X_n = 3125X_{n-1} \text{ modulo } (2^{35} - 31)$$

is a good choice.

The first number in the sequence (X_0) is called the *seed*. The seed completely determines the sequence of random numbers. In a good random generator, different seeds produce different sequences. Sometimes the computer selects the seed itself (e.g. based on a systems clock). However, preferably the user should consciously select a seed himself, allowing the *reproduction* of the simulation experiment later. Reproducing a simulation experiment is important whenever an unexpected phenomenon occurs that need further examination.

Most simulation languages and packages possess an adequate random generator. This generator can be seen as a black box: a device that produces (pseudo) random numbers upon request. However, beware: pseudo-random numbers are not truly random! (A deterministic algorithm is used to generate them.) Do not use more than one generator and take care in selecting the seed.
To illustrate the dangers in using random generators we mention two well-known pitfalls.
The first mistake is using the so-called 'lower order bits' of a random sequence. For example, if a random generator produces the number 0.1321734234, the higher order digits 0.13217 are 'more random' than the lower order digits 34234. In general the lower order digits show a clear cyclical behavior.
Another frequent mistake is the double use of a random number. Suppose that the same random number is used twice for generating a sample from a probability distribution. This introduces a dependency into the model that does not exist in reality, which may lead to extremely deceptive results.

3.2 Some distributions

Only rarely do we need random numbers uniformly distributed between 0 and 1. Depending on the situation, we need samples from different *probability distributions*. A probability distribution specifies which values are possible and how probable each of those values is.

Probability distribution

To simplify the discussion of random distributions and samples from probability distributions, we introduce the term *random variable*. A random variable X is a variable with a certain probability of taking on certain values. For example, we can model the throwing of a dice by means of a variable X that can take on the values 1,2,3,4,5 and 6. The probability of obtaining any value a from this set is $\frac{1}{6}$. We can write this

Random variable

as follows:

$$\mathbb{P}[X = a] = \begin{cases} \frac{1}{6} & \text{if } a \in \{1, 2, 3, 4, 5, 6\} \\ 0 & \text{else} \end{cases}$$

Expectation

Variance

$\mathbb{E}[X]$

$Var[X]$

Given a random variable X we can define its *expectation* and *variance*. The expectation of X, denoted by $\mathbb{E}[X]$, is the average to be expected from a large number of samples from X. We also say the *mean* of X. The variance, denoted as $\text{Var}[X]$, is a measure for the average deviation of the mean (expectation) of X. If X has a high variance, many samples will be distant from the mean. A low variance means that in general samples will be close to the mean. The expectation of a random variable X is often denoted with the letter μ, the variance ($\text{Var}[X]$) is denoted as σ^2. The relation between expectation and variance is defined by the following equality:

$$\text{Var}[X] = \mathbb{E}[(X - \mu)^2] = \mathbb{E}[X^2] - \mu^2$$

As $\text{Var}[X]$ is the expectation of the *square* of the deviation from the mean, the square root of $\text{Var}[X]$ is a better measure for the deviation from the mean. We call $\sigma = \sqrt{\text{Var}[X]}$ the *standard deviation* of X. If the standard deviation of X is large compared to the mean, we may say that X is *wildly* distributed. For more details we refer to Appendix A and the list of references.

Standard deviation

In this section we show some widely used probability distributions, starting with *discrete* probability distributions. The values that a discretely distributed random variable can take on have are separated; the distance between any two such values will exceed a certain minimum. Often, there are only finitely many such values.

3.2.1 Bernoulli distribution

If X is a random variable distributed according to a *Bernoulli distribution* with parameter p, then it may take on the value 1 with probability p and 0 with probability $1 - p$. So:

Bernoulli distribution

p

$$\mathbb{P}[X = 0] = 1 - p \text{ en } \mathbb{P}[X = 1] = p$$

The expectation $\mathbb{E}[X]$, i.e. the mean value, is p. The variance $\text{Var}[X]$ equals $p(1 - p)$. Often, the value 1 signifies success and 0 failure of an experiment.

3.2.2 Discrete homogeneous distribution

A random variable X is distributed according to a *discrete homogeneous distribution* with a lower bound of a and an upper bound of b, if it can

Homogeneous distribution

a, b

take on only integer values between and including a and b and each such value is equally probable. The lower bound a and the upper bound b are integers. In this case the probability of a certain value k equals:

$$\mathbb{P}[X = k] = \begin{cases} \frac{1}{(b-a)+1} & \text{if } k \in \{a, a+1, a+2, \ldots, b-1, b\} \\ 0 & \text{else} \end{cases}$$

The expectation ($\mathbb{E}[X]$) equals $\frac{a+b}{2}$. The variance ($\mathrm{Var}[X]$) equals $\frac{(b-a)(b-a+2)}{12}$. Rolling a dice corresponds to taking a sample from a discrete homogeneous distribution with lower bound 1 and upper bound 6.

3.2.3 Binomial distribution

Suppose we do n experiments that can either succeed or fail. Per experiment the chance of success equals p. What is the probability of k successes? We can model this with a *binomial distribution* with parameters n and p. Suppose X is distributed binomially with parameters n and p. For $x \in \{0, 1, \ldots, n\}$ we have:

Binomial distribution

n, p

$$\mathbb{P}[X = k] = \binom{n}{k} p^k (1-p)^{n-k}$$

The expectation equals np. The variance equals $np(1-p)$. Throwing 10 coins on a table and counting the tails corresponds to taking a sample from a binomial distribution with parameters $n = 10$ and $p = 0.5$. A special case of the binomial distribution is the Bernoulli distribution ($n = 1$).

3.2.4 Geometrical distribution

Suppose repeating an experiment with a chance of success p until we are successful for the first time. The number of experiments thus needed is a sample from a *geometrical distribution* with parameter p. Suppose X being distributed geometrically with parameter p, then for each positive integer number k:

Geometrical distribution

$$\mathbb{P}[X = k] = (1-p)^{k-1} p$$

The expectation equals $\frac{1}{p}$. The variance equals $\frac{1-p}{p^2}$. This distribution is also called the Pascal distribution.

3.2.5 Poisson distribution

The *Poisson distribution* is strongly related to the negative exponential distribution seen later in this book. If the time between two consecutive arrivals of a customer in a supermarket is distributed according to the negative exponential distribution with parameter λ, the number of

Poisson distribution

customers entering the supermarket per time unit is distributed Poisson with parameter λ.

λ

Suppose X is distributed Poisson with parameter λ, then for each integer number k the following holds:

$$\mathbb{P}[X = k] = e^{-\lambda}\frac{\lambda^k}{k!}$$

The expectation ($\mathbb{E}[X]$) equals λ. The variance ($\mathrm{Var}[X]$) also equals λ. Think of a supermarket where customers enter according to a negative exponential distribution. The average time between two arrivals is 15 minutes (0.25 hour), that is to say $\lambda = \frac{1}{0.25} = 4$. The number of arrivals per hour is distributed Poisson with parameter $\lambda = \frac{1}{0.25} = 4$. The average number of arrivals per hour therefore equals 4. The variance also equals 4.

Next in this section we will discuss some continuous distributions. Unlike discrete distributions, the notion $\mathbb{P}[X = k]$ giving the probability of a certain value k has no importance. (For continuous distributions, $\mathbb{P}[X = k] = 0$ for any k.) The important notion is the *probability density*. The larger the probability density $f_X(k)$ of a continuously distributed random variable X at the point k becomes, the greater the probability that a sample from X is close to k.

Probability density

3.2.6 Uniform distribution

The *uniform distribution*, also called homogeneous distribution, is very simple. Between a lower bound a and an upper bound b all values are equally probable. A random variable X distributed uniformly with parameters a and b has probability density $f_X(x) = \frac{1}{b-a}$ for x between a and b ($a \leq x \leq b$) and $f_X(x) = 0$ for $x < a$ or $x > b$. Figure 3 is a graph showing the probability density for the uniform distribution with $a = 2$ and $b = 4$.

Uniform distribution

a, b

From this figure we infer that all values between a and b are equally probable. The expectation ($\mathbb{E}[X]$) equals $\frac{a+b}{2}$ The variance ($\mathrm{Var}[X]$) equals $\frac{(b-a)^2}{12}$.

3.2.7 Negative exponential distribution

The *negative exponential distribution* is often used to model arrival processes. The negative exponential distribution has only one parameter λ. Let X be a negative exponentially distributed random variable with parameter λ. The corresponding probability density will be ($x \geq 0$):

Negative exponential distribution

λ

$$f_X(x) = \lambda e^{-\lambda x}$$

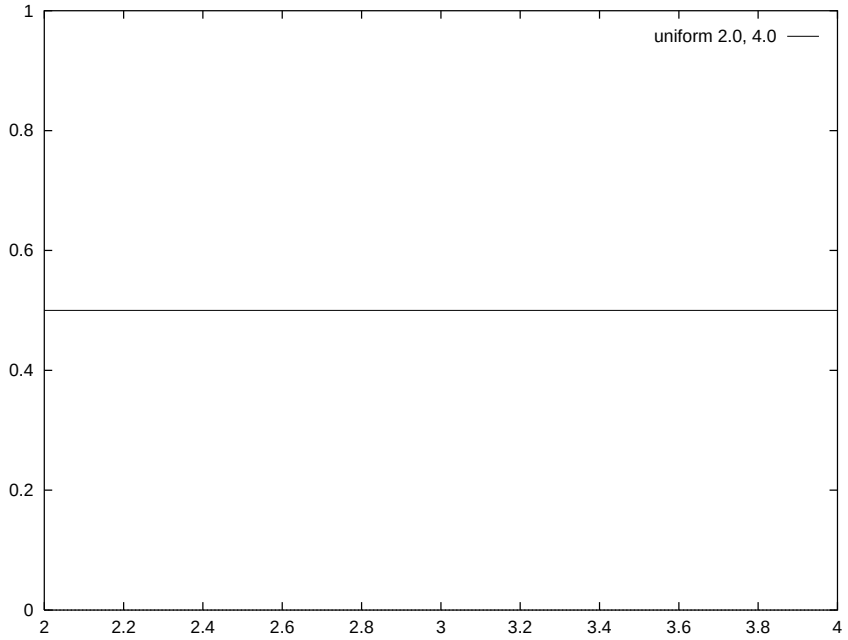

Figure 3: The uniform distribution with parameters $a = 2$ and $b = 4$.

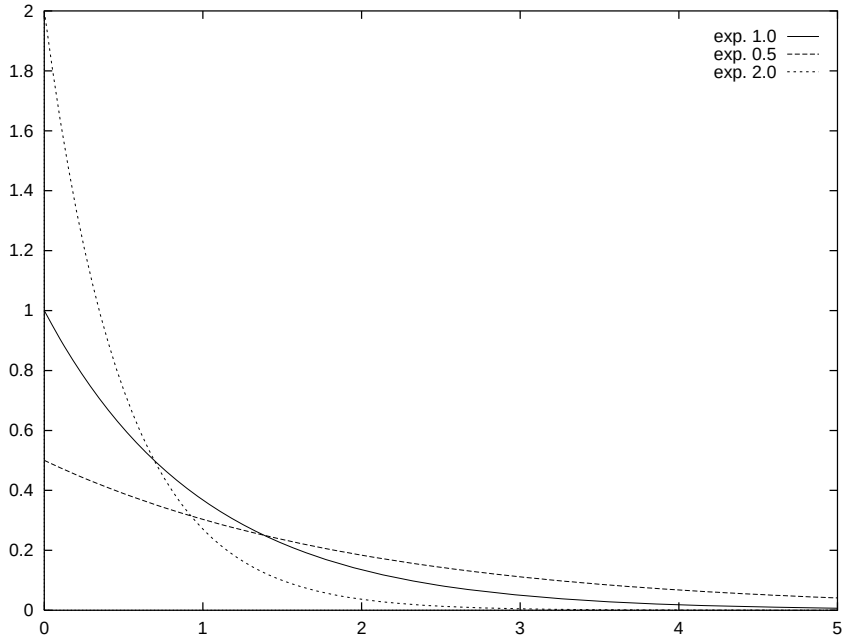

Figure 4: Negative exponential distributions with parameters $\lambda = 1.0$, $\lambda = 0.5$ and $\lambda = 2.0$.

Figure 4 shows this probability density for $\lambda = 1.0$, $\lambda = 0.5$ and $\lambda = 2.0$. The expectation equals $\frac{1}{\lambda}$. The variance equals $\frac{1}{\lambda^2}$. If a random variable X, negative exponentially distributed with parameter λ, defines an arrival process, λ is called the *intensity* of the arrival process. The random variable X is used to model the time between two consecutive arrivals. The higher the intensity of the arrival process (i.e. the larger λ), the larger the mean number of arrivals per time unit. If $\lambda = 10$, the expected time between two consecutive arrivals is 0.10 time units. The mean number of arrivals per time unit in this case equals 10. If $\lambda = 100$, then the expected time between two consecutive arrivals equals 0.01 time units and the mean number of arrivals per time unit equals 100.

Intensity

3.2.8 Normal distribution

The *normal distribution* has many applications. This distribution is used for modeling processing times, response times, stock levels and transport times. Often, one does not know the exact distribution of a certain random variable, but some of its characteristics (like mean and variance) are known. In such cases, the normal distribution is often used as an approximation. This standard practice is however not without dangers; it needs justification! The normal distribution has two parameters μ and σ. The probability density of a σ, μ-normally distributed random variable X is defined as follows:

Normal distribution

μ, σ

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Unlike the negative exponential distribution this random variable can also take on negative values. Figure 5 shows for a number of parameter values the corresponding probability densities. The probability densities are maximal around μ. As we digress from the mean, the probability density decreases. The smaller σ, the faster the probability density decreases, so for small σ, a value near μ is very likely. The expectation ($\mathbb{E}[X]$) equals μ. The variance ($\mathrm{Var}[X]$) equals σ^2.
If $\mu = 0$ and $\sigma = 1$, we call this a *standard normal distribution*. If Y is a standard normally distributed random variable, then $X = \mu + \sigma Y$ is a normally distributed random variable with parameters μ and σ. Conversely, if X is a normally distributed random variable with parameters μ and σ, then $Y = \frac{X-\mu}{\sigma}$ is a standard normally distributed random variable.

Standard normal distribution

If we use a normally distributed random variable for modeling time durations, like processing times, response times or transport times, we must be aware that this random variable can also take on *negative* values. In general negative durations are impossible; this may even cause a failure of the simulation software. To circumvent this problem, we might take a

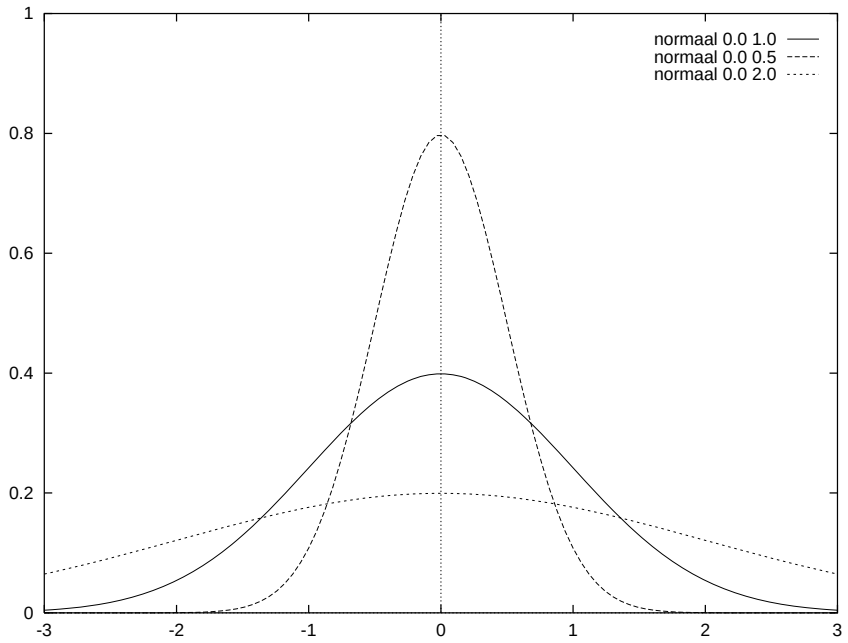

Figure 5: Normal distributions with parameters $\mu = 0.0$ and σ consecutively 1, 0.5 and 2.

new sample whenever the given sample produces a negative value. Note that this will affect the mean and the variance. Therefore, this solution is recommended only if the probability of a negative value is very small. We use the following rule of thumb: "If $\mu - 2\sigma < 0$ we cannot omit negative values generated.".

3.2.9 Gamma distribution

A characteristic of the normal distribution is its symmetry. We often need 'skewed' distributions instead, where the probability densities below and above the mean are distributed differently. In this case a *gamma distribution* might be chosen. A random variable X is gamma distributed with parameters $r > 0$ and $\lambda > 0$ if for all $x > 0$:

Gamma distribution

r, λ

$$f_X(x) = \frac{\lambda(\lambda x)^{r-1}\ e^{-\lambda x}}{\Gamma(r)}$$

(The function Γ is the mathematical gamma function.) Figures 6, 7 and 8 show the probability densities for a number of parameter values.
These figures clearly show that the gamma distribution, depending on the parameter values has many manifestations. If r does not exceed 1, f_X is a monotonously decreasing function. In this case values close to 0 are most probable. If $r > 1$, the function will first grow to a maximum and then decrease monotonously. The parameter r therefore determines the shape of the distribution. The parameter λ determines the 'spread'

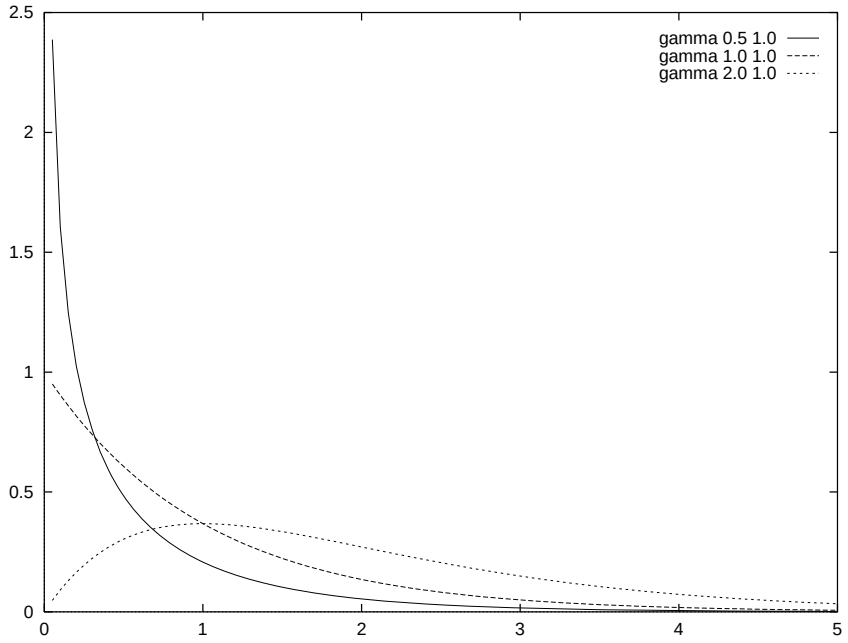

Figure 6: Gamma distributions with parameter $\lambda = 1$ and r consecutively 1, 0.5 and 2.

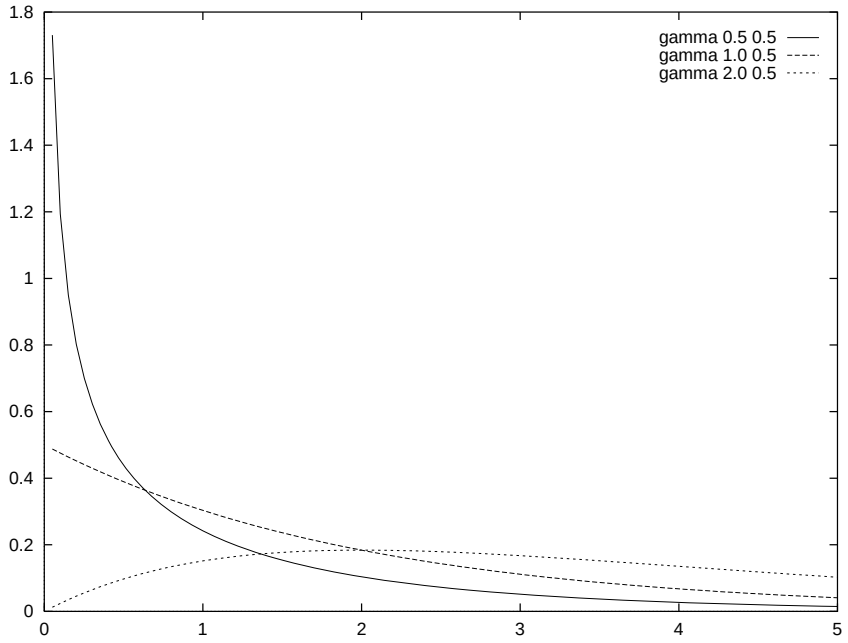

Figure 7: Gamma distributions with parameter $\lambda = 0.5$ and r consecutively 1, 0.5 and 2.

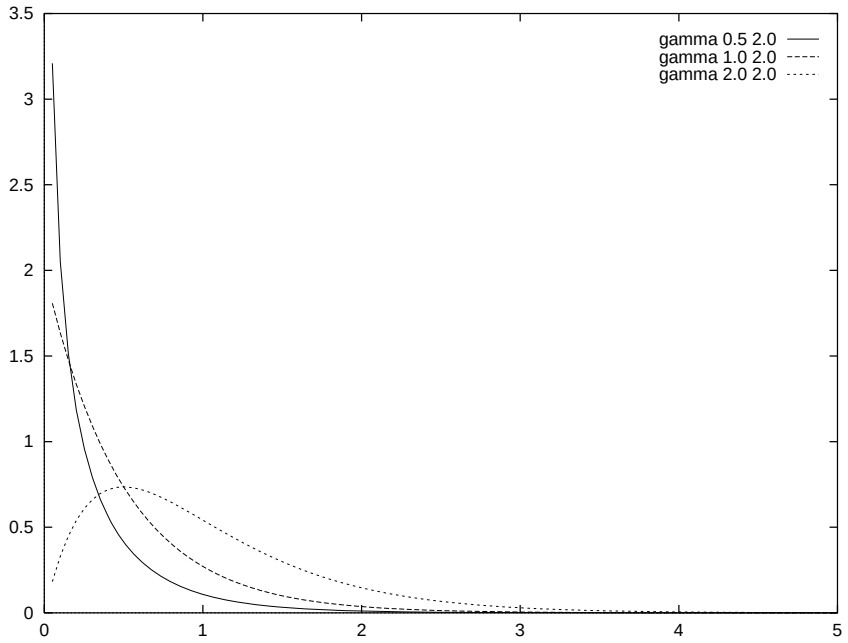

Figure 8: Gamma distributions with parameter $\lambda = 2$ and r consecutively 1, 0.5 and 2.

of the distribution. The larger λ, the more probable is a value close to 0. The expectation ($\mathbb{E}[X]$) equals $\frac{r}{\lambda}$. The variance ($\mathrm{Var}[X]$) equals $\frac{r}{\lambda^2}$.

Modal value

The *modal value* of a probability distribution is the value for which the probability density is maximal, i.e. m is the modal value of an X if for each x: $f_X(m) \geq f_X(x)$. The modal value of a gamma distributed random variable X depends on r and λ. If $r \leq 1$, 0 is the most probable value, i.e. the modal value of X equals 0. If $r > 1$ the modal value equals $\frac{r-1}{\lambda}$. A special case of gamma distribution is the negative exponential distribution ($r = 1$).

3.2.10 Erlang distribution

Erlang distribution

The *Erlang distribution* is a special case of the gamma distribution. A gamma distribution with an integer parameter r, is also called an Erlang distribution. A random variable which is distributed Erlang with parameters r (integer) and λ can be considered the sum of r independent, negative exponentially distributed random variables with parameters λ. If X is distributed Erlang with parameters r and λ, then:

r, λ

$$f_X(x) = \frac{\lambda(\lambda x)^{r-1}\ e^{-\lambda x}}{(r-1)!}$$

(Note that $n!$ is the factorial function: the product of all integers from 1 up to n. If r is an integer, then $\Gamma(r) = (r-1)!$.) Figure 9 shows for a

number of values of r and λ the function f_X.

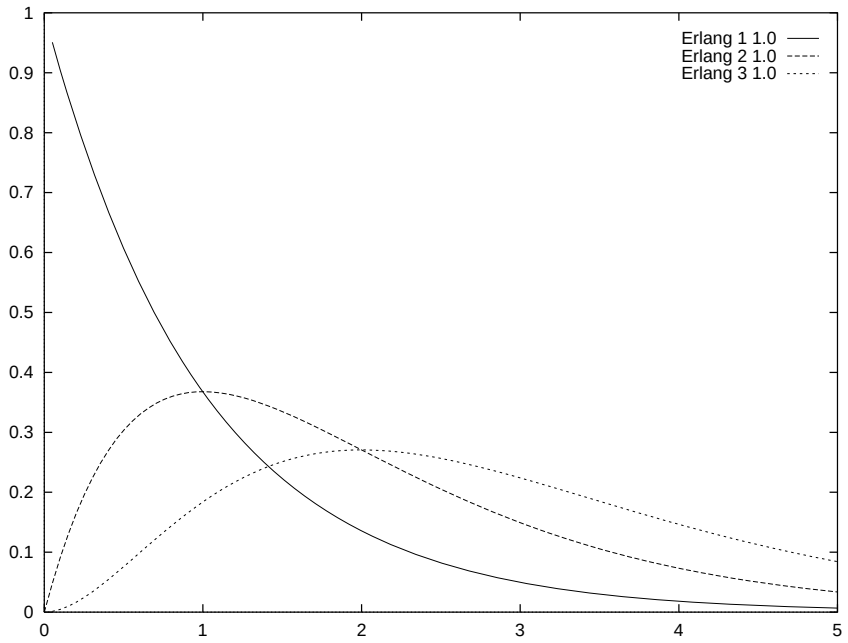

Figure 9: Erlang distributions with parameter $\lambda = 2$ and r consecutively 1, 2 and 3.

The expectation equals $\frac{r}{\lambda}$. The variance equals $\frac{r}{\lambda^2}$.

3.2.11 χ^2 distribution

χ^2 *distribution*

Degrees of freedom

The χ^2 *distribution* is another special case of the gamma distribution. A χ^2 distribution has a single parameter v. This v is a positive integer and represents the number of *degrees of freedom*. A random variable X distributed χ^2 with v degrees of freedom is the same as a random variable distributed gamma with parameters $r = \frac{v}{2}$ and $\lambda = \frac{1}{2}$. Figure 10 shows the probability density for a number of values v. The expectation equals v. The variance equals $2v$. There is also a connection between the normal distribution and the χ^2 distribution. If $X_1, X_2, \ldots, X_n$ are mutually independent standard normally distributed random variables, the random variable $X = X_1^2 + X_2^2 + \ldots + X_n^2$ is distributed exactly χ^2 with parameters $v = n$. The χ^2 distribution is used specifically for 'goodness-of-fit' tests.

3.2.12 Beta distribution

Beta distribution

a, b, r, s

Like the uniform distribution, the *beta distribution* is distributed over a finite interval. We use it for random variables having a clear upper and lower bound. The beta distribution has four parameters a, b, r and s. The parameters a and b represent the upper and lower bounds of

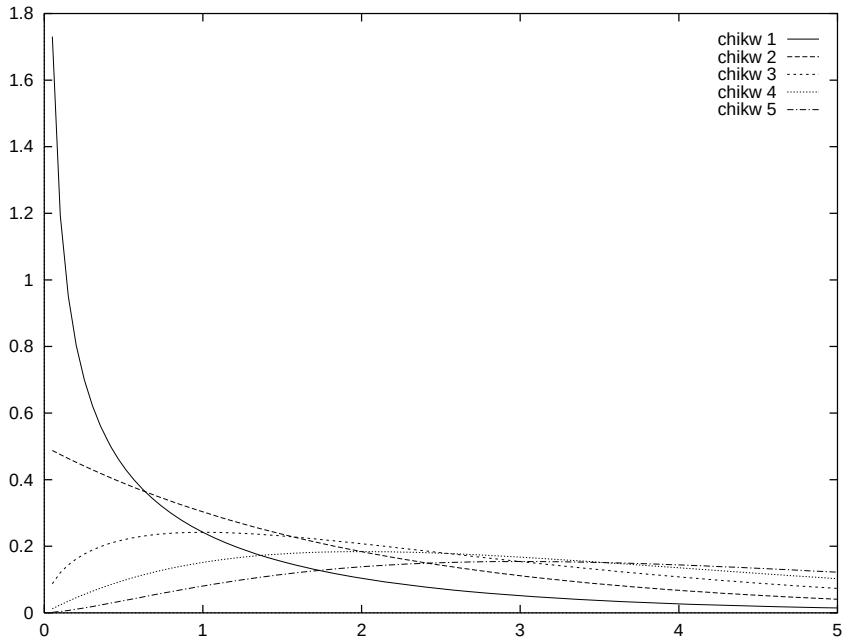

Figure 10: χ^2 distributions with $v = 1$, $v = 2$, $v = 3$, $v = 4$ and $v = 5$.

the distribution. The parameters r ($r > 0$) and s ($s > 0$) determine the shape of the distribution. For each x ($a \leq x \leq b$), the probability density is defined as follows:

$$f_X(x) = \frac{1}{b-a} \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} \left(\frac{x-a}{b-a}\right)^{r-1} \left(\frac{b-x}{b-a}\right)^{s-1}$$

Standard beta distribution

If $a = 0$ and $b = 1$ we call this a *standard beta distribution*. Figures 11, 12 and 13 show, assuming a standard beta distribution, for a number of values of r and s the corresponding probability densities. Clearly, the beta distribution is very varied.
If $r = s = 1$, X is distributed homogeneously with parameters a and b. The larger/smaller $\frac{r}{s}$ becomes, the greater/lesser the chance of a value close to b. If $r < 1$, the probability density is large close to a. If $s < 1$, the probability density is large close to b. If $r > 1$ and $s > 1$, the modal value (the maximum probability density) lies somewhere between a and b. The expectation equals:

$$\mathbb{E}[X] = a + (b-a)\frac{r}{r+s}$$

The variance equals:

$$\mathrm{Var}[X] = \frac{rs(b-a)^2}{(r+s)^2(r+s+1)}$$

A beta distribution is 'skewed' (not symmetrical) whenever $r \neq s$. Therefore, the most probable value (modal value) may differ from the expectation. If $r > 1$ and $s > 1$, the modal value equals $a + (b-a)\frac{r-1}{r+s-2}$. If

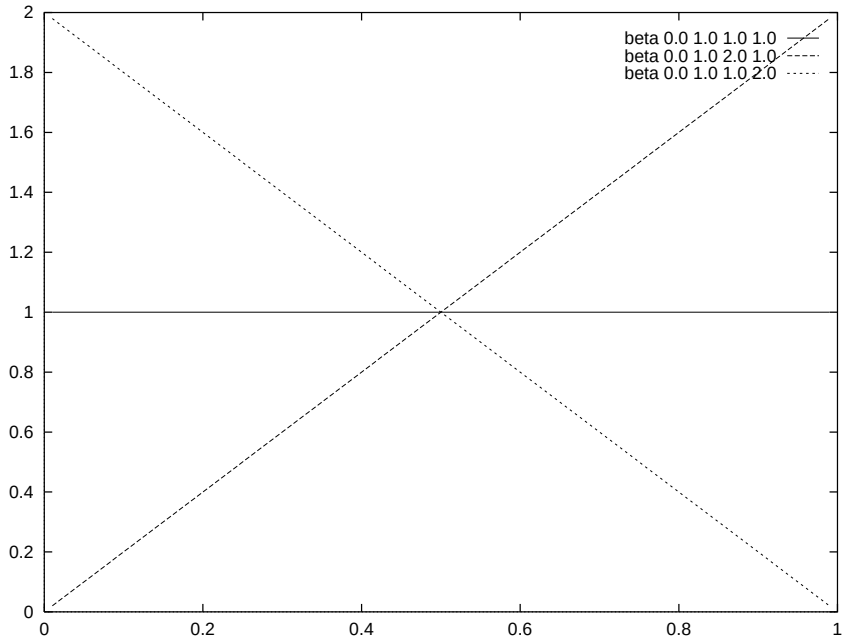

Figure 11: Beta distributions with parameters $a = 0$, $b = 1$ and r consecutively 1.0, 2.0 and 1.0, and s consecutively 1.0, 1.0 and 2.0.

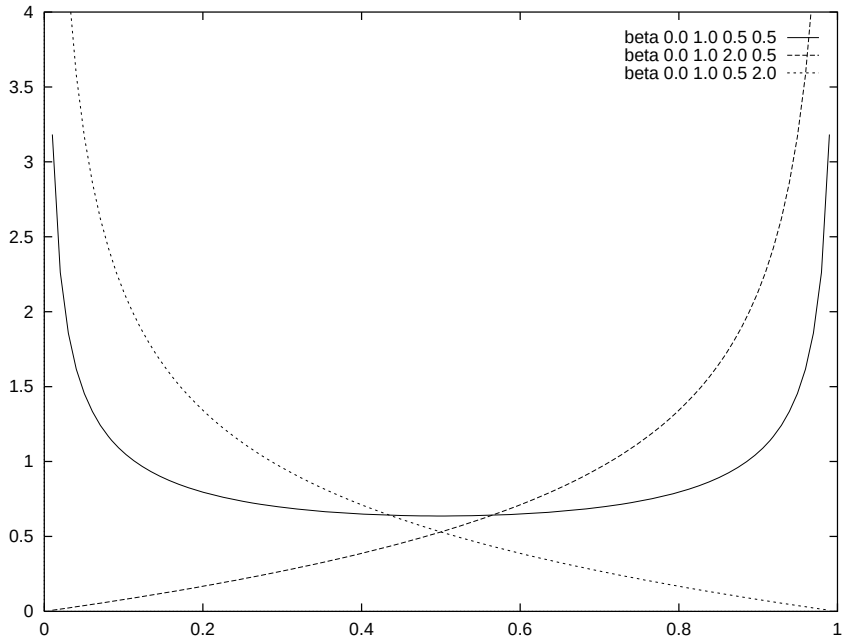

Figure 12: Beta distributions with parameters $a = 0$, $b = 1$ and r consecutively 0.5, 2.0 and 0.5, and s consecutively 0.5, 0.5 and 2.0.

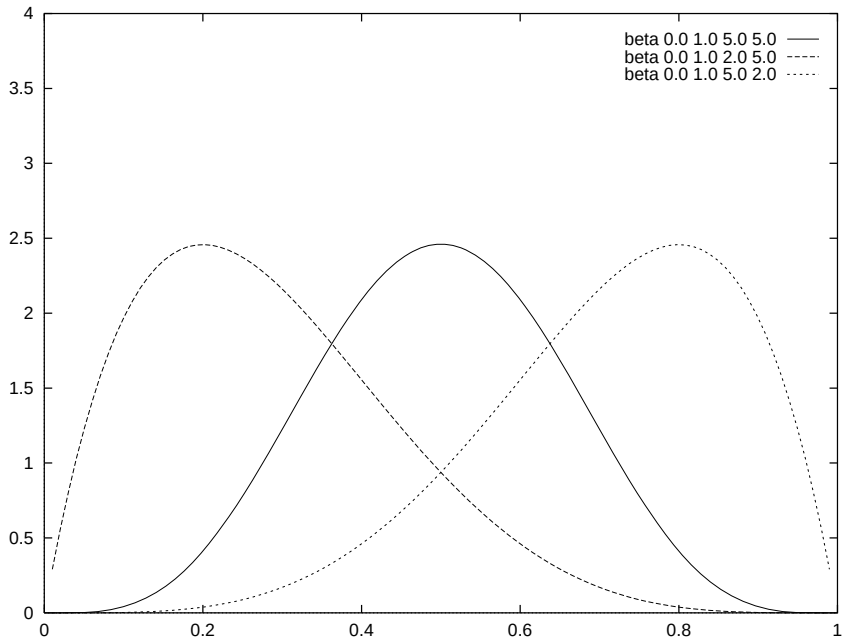

Figure 13: Beta distributions with parameters $a = 0$, $b = 1$ and r consecutively 5.0, 2.0 and 5.0, and s consecutively 5.0, 5.0 and 2.0.

$r < 1$ or $s < 1$, the maximal probability density is achieved at b (if $r < s$) or a (if $s < r$).
Suppose we want a standard beta distribution with an expectation μ and a variance σ^2. In that case we have to choose r and s as follows:

$$\begin{aligned} r &= \frac{\mu^2(1-\mu)}{\sigma^2} - \mu \\ s &= \frac{(1-\mu)^2\mu}{\sigma^2} - (1-\mu) \end{aligned}$$

This is only possible if the following condition is met:

$$\sigma^2 < \mu(1-\mu)$$

As each sample is between 0 and 1, there is an upper bound to the variance (at most 0.25).
A beta distributed random variable Y with lower bound a, upper bound b, expectation μ and variance σ^2, is obtained by constructing a random variable X, distributed standard beta with expectation $\frac{\mu-a}{b-a}$ and variance $\frac{\sigma^2}{(b-a)^2}$. Y is then defined by $Y = (b-a)X + a$.

PERT

A well-known application of the beta distribution is *PERT* (Program Evaluation and Review Technique). PERT is a technique used by project managers assess throughput times. For each activity PERT needs three estimates of its duration:

(i) an optimistic estimate (i.e. a lower bound),

(ii) a pessimistic estimate (i.e. an upper bound),

(iii) an estimate of the most probable duration (modal value).

If we model the duration of such an activity by means of a beta distribution, we use the first two estimates to determine the parameters a and b. If c is the most probable duration, the parameters r and s are set so that the expectation and the variance take on the following values:

$$\begin{aligned} \mu &= \frac{a+4c+b}{6} \\ \sigma^2 &= \frac{(b-a)^2}{36} \end{aligned}$$

Therefore, if a lower and upper bound are given, the variance is fixed.

4 Processing the results

We use simulation to assess present or future situations. During simulation, measurements are taken. Suppose, a bank considers buying new cash dispensers. Through simulation, information must be obtained about waiting times and errors. In this section we show how assertions can be made about these quantities from a simulation study.

4.1 Mean and variance

Observations

During simulation there are repeated *observations* of quantities such as waiting times, run times, processing times, and/or stock levels.

`Example`

A player throws a dice 10 times and observes the number that comes up. The following observations are made : 3, 4, 6, 1, 1, 2, 5, 3, 3, and 2.

Separate observations have little interest. People use simulation to obtain statistical information about a certain quantity, e.g. to assess the average throughput time of an order.

Suppose we have k consecutive observations, called $x_1, x_2, ...x_k$. These observations are also called a *random sample*.

Sample mean

The *mean* of a number of observations is also called the *sample mean*. We denote the sample mean of $x_1, x_2, ...x_k$ as $\overline{x}$. We can find $\overline{x}$ by adding the observations and dividing the sum by k:

$$\overline{x} = \frac{\sum_{i=1}^{k} x_i}{k}$$

Please note that the sample mean is an estimate of the true mean.

Sample variance

The *variance* of a number of observations is also called the *sample variance*. This variance is a measure for the deviation from the mean. The

smaller the variance, the closer the observations will be to the mean. We can find the sample variance s^2 by using the following formula:

$$s^2 = \frac{\sum_{i=1}^{k}(x_i - \overline{x})^2}{k-1}$$

We can rewrite this formula as:

$$s^2 = \frac{(\sum_{i=1}^{k} {x_i}^2) - \frac{1}{k}\left(\sum_{i=1}^{k} x_i\right)^2}{k-1}$$

If during a simulation we keep track of (1) the number of observations k, (2) the sum of all observations $\sum_{i=1}^{k} x_i$ and (3) the sum of the squares of all observations $\sum_{i=1}^{k} {x_i}^2$, we can determine the sample mean and the sample variance. The square root of the sample variance $s = \sqrt{s^2}$ is also called the *sample standard deviation*. The standard deviation s gives a more adequate impression of the deviations from the mean than the sample variance s^2.

Sample standard deviation

Apart for the mean and the variance of a random sample, there is also the so-called *median* of k observations $x_1, x_2, \ldots, x_k$. The median is the value of the observation in the 'middle' after sorting the observations w.r.t. their value. An 'observation in the middle' only exists if the number of observations is odd. In case of an even number of observations, the average of the two observations in the middle is taken. In a simulation experiment, we have to save and sort all observations in order to calculate their median.

Median

`Example`

In a simulation experiment the following waiting times are measured: 2.3, 3.4, 2.2, 2.8, 5.6, 3.2, 6.8, 3.2, 5.3 and 2.1. Using the random sample we can determine the sample mean and the sample variance. The sample mean is 3.69 and the sample variance is 2.648. The median is 3.2.

4.2 Subruns and preliminary run

In a simulation experiment, we can easily determine the sample mean and the sample variance of a certain quantity. We can use the sample mean as an estimate for the expected true value of this quantity (e.g. waiting time), but we can not determine how reliable this estimate is. A simulation experiment consists of a number of partial experiments (subruns) that allow us to assess the reliability of the simulation results.

4.2.1 Necessity

`Example`

Consider a post office with one counter. Customers enter the post office according to a Poisson process with intensity 6. The time between two consecutive customers is therefore distributed negative exponentially

with an average of 0.167 hours (10 minutes). The service time is also distributed negative exponentially. The average service time is 0.1 hours (6 minutes). We want to determine the average waiting time of customers. Based on a simulation experiment that assesses the waiting times of 1000 consecutive customers, we observe a mean waiting time of 0.21 hours (approximately 12 minutes). (These are the results of a truly conducted simulation experiment.) The expected waiting time can be calculated mathematically in this case. (M/M/1-queue, see appendix C.) This calculated mean waiting time equals 0.15 hours (9 minutes). Conducting a longer simulation of, say, 50.000 customers will corroborate this last value. The estimate based on the first simulation experiment therefore differs substantially from the real expected waiting time.

Therefore, we need a mechanism to determine the reliability of a certain result. One might think that the sample variance can be used to answer this question. However, this is not the case because the sample variance based on the waiting times of 1000 consecutive customers only estimates the average deviation from the sample mean waiting time. It does not say how well the sample mean estimates the expectation. We tackle this problem by introducing *subruns*. Instead of one long simulation run we conduct a series of smaller simulation runs, from which we compare the results.

Subruns

Suppose that instead of one long simulation run with 1000 customers, we had executed 10 consecutive runs with 100 customers each. Table 1 gives for each of these subruns a separate sample mean. The mean over these 10 subruns is about 0.21 hours. The table clearly exhibits the differences between the various sample means. The estimated waiting time varies between 0.12 and 0.31 hours. Therefore, these data are not sufficient to estimate the waiting time to be expected. So, by dividing the simulation experiment into subruns, we obtained an impression of the reliability. We can also quantify this impression.

4.2.2 Subruns and initial phenomena

Instead of one long simulation run, we use several smaller subruns. In doing so, we need to distinguish between two situations:

Stable situation

(i) A *stable situation*, meaning the circumstances are constant. Quantities like e.g. the arrival intensities of customers are the same throughout the simulation. There are no structural peaks. Also, no startup effects occur: we are interested in the *stable state* of a process.

Unstable situation

(ii) An *unstable situation*, where the circumstances change structurally during simulation. For instance, an "empty" state characterizes the

subrun number	mean waiting time
1	0.16
2	0.14
3	0.29
4	0.12
5	0.26
6	0.31
7	0.13
8	0.28
9	0.21
10	0.28

Table 1: The sample mean itemized per subrun.

beginning and end of the simulation. Often there is a clear start and end; this is called a *terminating process*.

Analyzing the average waiting time of a customer in the post office at a given time of the day can be done in stable situation. We assume that the arrival process and the service process have certain characteristics at this time of the day. However, the analysis of the average waiting time of customers during the whole day requires an unstable situation. We might see a different arrival process at the end of the day than at 10 AM. The reason that we differentiate between stable and unstable situations is the fact that this influences the format of the simulation experiment.

In case of a stable situation

If we are dealing with a stable situation, we can execute the subruns sequentially. Each subrun, except the first, starts in the final state of the last subrun. This means that we can execute one long simulation run and cut it into equal pieces (subruns). We just store the relevant results of each subrun. We have to treat the first piece of such a simulation run carefully. For example, the choice of the seed influences the first measured results. Also, so-called *initial phenomena* may cause the results to be influenced by a chosen initial state. If we start with an empty post office, the first arriving customer will be immediately served. While this is a limited effect, there are situations where the initial state chosen will have a long-lasting influence on certain quantities. If, in simulating a manufacturing plant, we choose an empty initial state (i.e., with no work at hand) if will take some time before the throughput times measured do reflect the corresponding quantities during normal operation.

Initial phenomena

To minimize the effect of initial phenomena we start with a special subrun called the preliminary run (or start-up run). The results gathered

in this run will be disregarded when estimating a certain quantity. The preliminary run has to be long enough for the initial phenomena to disappear. When simulating our manufacturing plant we can conclude the preliminary run as soon as the amount of work at hand has reached a stable level.

In case of an unstable situation

In dealing with an unstable situation, we can no longer execute the subruns after one another. Initial phenomena, if any, are an essential part of the simulation. Each subrun therefore has to start from the same starting condition. By using a different seed for each subrun, the results of the various subruns are still independent and can be compared to each other during analysis. If we want to analyze the waiting times in a post office for the entire day, each subrun represents one day. Each of these subruns starts in the state that there are no customers in the post office and ends after, at the end of the day, all customers have left.

Relevant questions when setting up a simulation experiment are:

- Is a preliminary run necessary?
- How long should the preliminary run be?
- How many subruns are necessary?
- How long should each subrun be?

Is a preliminary run necessary?

How long should the preliminary run be?

How many subruns are necessary?

How long should each subrun be?

Let us try and answer these questions. A preliminary run is necessary only if we are simulating a stable situation, where the initial state differs from the average state experienced during simulation. The length of a preliminary run depends on how much the initial state differs from an average state and how fast the simulation reaches a stable situation. We can chart the development of some relevant quantities in time and estimate from this chart when a stable state has been reached. This question is a difficult one. For example, the required number of subruns strongly depends on the desired reliability of the final results and the length of each subrun. We will deal with this subject later. For unstable situations, the length of a subrun is usually fixed. The length is e.g. one day, one morning or the time needed to handle one request. For stable situations, we must make certain that each subrun is long enough to ensure that the initial state of one subrun does not depend upon the initial state of the next subrun. Suppose that at the start of a subrun there is an extremely long queue. If the length of a subrun is too short, this queue will not have been processed in the next subrun. In that case a dependency exists between the results of the two subruns. This can lead to an incorrect interpretation of results. A rule of thumb is that each subrun should contain at least one regeneration point. A regeneration

point is a certain state that the system regularly returns to. An example regeneration point for the post office simulation is the state where the post office is empty of customers.

4.3 Analysis of subruns

Suppose we have executed n subruns and measured a certain result x_i for each subrun i. We know there exists a "true" value (called μ) that each x_i approximates and we want to derive assertions about μ from the x_i. For example, x_i is the mean waiting time measured in subrun i and μ the "true" mean waiting time that we could find by conducting a hypothetical simulation experiment of infinite length. (Instead, we might consider the mean variance of the waiting time, the mean occupation rate of a server or the mean length of a queue.) We must be certain that the values x_i are mutually independent for all subruns. (We have, e.g. forced this by choosing a long enough subrun length). Given the results $x_1, x_2, \ldots, x_n$, we derive the sample mean $\overline{x}$:

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

and the sample variance s^2:

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n-1}$$

Please note!

Note that the sample mean and the sample variance for the results of the various subruns should not be confused with the mean and the variance of a number of measures *within* one subrun! We can consider $\overline{x}$ as an estimate of μ. The value $\overline{x}$ can be seen as a sample from a random variable $\overline{X}$ called *estimator*. The value [1] $\frac{s}{\sqrt{n}}$ is an indication of the reliability of the estimate $\overline{x}$. If $\frac{s}{\sqrt{n}}$ is small, it is a good estimate.

4.3.1 The situation with over 30 subruns

If there is a large number of subruns, we can consider the estimator $\overline{X}$ (because of the central limit theorem) as normally distributed. We will therefore treat the situation with over 30 subruns as a special case.

When do we stop generating subruns?

The fact that $\frac{s}{\sqrt{n}}$ measures how well $\overline{x}$ approximates μ, allows us to determine the time that we can stop generating subruns.

(i) Choose a value d for the permitted standard deviation from the estimated value $\overline{x}$.

[1] The variance of the estimator $\overline{X}$ is $\mathrm{Var}[\overline{x}] = \mathrm{Var}[\frac{1}{n}\sum_{i=1}^{n} x_i] = \frac{\sigma^2}{n}$. The standard deviation of $\overline{x}$ thus equals $\frac{\sigma}{\sqrt{n}}$. As s is a good estimate of σ, the amount $\frac{s}{\sqrt{n}}$ yields a good estimate for the standard deviation of $\overline{x}$.

(ii) Generate at least 30 subruns and note per subrun the value x_i.

(iii) Generate additional subruns until $\frac{s}{\sqrt{n}} \leq d$, where s is the sample standard deviation and n the number of subruns executed.

(iv) The sample mean $\overline{x}$ is now an estimate of the quantity to be studied.

There are two other reasons why in this case at least 30 subruns have to be executed. In the first place, $\overline{X}$ is only approximately normally distributed with a large number of subruns. This is a compelling reason to make sure that there are at least 30 mutually independent subruns. Another reason for choosing an adequate number of subruns is the fact that by increasing the number of subruns, s becomes a better estimate of the true standard deviation.

Confidence interval

Given a large number of independent subruns, we can also determine a *confidence interval* for the quantity to be studied. Because $\overline{x}$ is the average of a large number of independent measures, we can assume that $\overline{x}$ is approximately normally distributed. (see Appendix A). From this fact, we deduce the probability that μ lies within a so-called confidence interval. Given the sample mean $\overline{x}$ and the sample standard deviation s, the true value μ conforms with confidence $(1-\alpha)$ to the following equation:

$$\overline{x} - \frac{s}{\sqrt{n}}\, z(\frac{\alpha}{2}) < \mu < \overline{x} + \frac{s}{\sqrt{n}}\, z(\frac{\alpha}{2})$$

where $z(\frac{\alpha}{2})$ is defined as follows. If Z is a standard normally distributed random variable, then $\mathbb{P}[Z > z(x)] = x$. For a number of values of x, $z(x)$ is shown in table 2. The value α represents the unreliability, that is the chance that μ does not conform to the equation. Typical values for α range from 0.001 to 0.100. The interval

$$\left[\overline{x} - \frac{s}{\sqrt{n}}\, z(\frac{\alpha}{2}), \overline{x} + \frac{s}{\sqrt{n}}\, z(\frac{\alpha}{2})\right]$$

is also called the $(1-\alpha)$-confidence interval for the estimated value μ.

x	z(x)
0.001	3.090
0.005	2.576
0.010	2.326
0.050	1.645
0.100	1.282

Table 2: $\mathbb{P}[Z > z(x)] = x$ where Z is standard normally distributed.

Example

We can illustrate the above with the following example. A company is worried about the workload of the help desk staff. This has become so high that absenteeism has increased substantially. To look into this situation a simulation study was done to determine how to decrease the workload. To assess the workload in the present situation, a simulation experiment consisting of 30 subruns was conducted. Each subrun represents one working day. The average occupation rate of help desk staff per subrun is shown in Table 3.

subrun number	average load factor	subrun number	average load factor	subrun number	average load factor
1	0.914	11	0.894	21	0.898
2	0.964	12	0.962	22	0.912
3	0.934	13	0.973	23	0.943
4	0.978	14	0.984	24	0.953
5	0.912	15	0.923	25	0.923
6	0.956	16	0.932	26	0.914
7	0.958	17	0.967	27	0.923
8	0.934	18	0.924	28	0.936
9	0.978	19	0.945	29	0.945
10	0.976	20	0.936	30	0.934

Table 3: De The average occupation rate per subrun.

The sample mean is 0.9408 and the sample variance is 0.000617. So, all the data needed to set up a $(1-\alpha)$-confidence interval are known: $n = 30$, $\overline{x} = 0.9408$, $s^2 = 0.000617$ and therefore $s = 0.02485$. If we take α equal to 0.010 we will find the following confidence interval:

$$\left[0.9408 - \frac{0.02485}{\sqrt{30}}\ z(\frac{0.010}{2}), 0.9408 + \frac{0.02485}{\sqrt{30}}\ z(\frac{0.010}{2})\right]$$

As $z(0.005) = 2.576$ this is therefore the interval $[0.9291, 0.9525]$. The larger the unreliability α, the smaller the corresponding confidence interval. For example, for $\alpha = 0.10$ we will find the confidence interval $[0.9333, 0.9483]$. From the results we can safely infer that the occupation rate for the help desk staff is quite high!

4.3.2 The situation with less than 30 subruns

In some cases we can do with less than 30 subruns. In this case, the results of the separate subruns (x_i) have to be approximately normally

distributed. If the result of a subrun x_i is the average of a large number of observations, then (by the central limit theorem,) each x_i is approximately normally distributed. So if x_i is, say, the average waiting time, average service time or average throughput time of a large number of customers, x_i is approximately normally distributed. By using this property, we can deduce, given n subruns with a sample mean $\overline{x}$, sample deviation s and reliability $(1-\alpha)$ the following confidence interval:

$$\left[\overline{x} - \frac{s}{\sqrt{n}}\ t_{n-1}(\frac{\alpha}{2}), \overline{x} + \frac{s}{\sqrt{n}}\ t_{n-1}(\frac{\alpha}{2})\right]$$

Student's t-distribution

where $t_v(x)$ is the critical value of a *Student's t-distribution*, also called t-distribution, with v degrees of freedom. Table 4 shows for a number of values of v and x the critical value $t_v(x)$.

$t_v(x)$	$x =$			
	0.100	0.050	0.010	0.001
$v = 1$	3.08	6.31	31.82	318.31
2	1.89	2.92	6.96	22.33
3	1.64	2.35	4.54	10.21
4	1.53	2.13	3.75	7.17
5	1.48	2.02	3.37	5.89
6	1.44	1.94	3.14	5.21
7	1.41	1.89	3.00	4.79
8	1.40	1.86	2.90	4.50
9	1.38	1.83	2.82	4.30
10	1.37	1.81	2.76	4.14
15	1.34	1.75	2.60	3.73
20	1.33	1.72	2.53	3.55
25	1.32	1.71	2.49	3.45
50	1.30	1.68	2.40	3.26
100	1.29	1.66	2.35	3.17
∞	1.28	1.64	2.33	3.09

Table 4: The critical values for a Student's t-distribution with v degrees of freedom.

Contrary to the method discussed earlier, we can determine the confidence interval in the way shown above if a limited number of subruns (say 10) is at our disposal. If we have a larger number of subruns at our disposal, it is better to apply the $(1-\alpha)$ confidence interval mentioned earlier, even if we are convinced that the subrun results are normally distributed. For large n the confidence interval based on $t_{n-1}(\frac{\alpha}{2})$ is more

accurate than the one based on $z(\frac{\alpha}{2})$; only the latter depends upon the central limit theorem concerning the number of subruns.

For any assertion concerning the reliability of $\overline{x}$, based on the results in table 2 we will find for $\alpha = 0.100$ the following confidence interval:

$$\left[0.9408 - \frac{0.02485}{\sqrt{30}}\ t_{29}(\frac{0.100}{2}), 0.9408 + \frac{0.02485}{\sqrt{30}}\ t_{29}(\frac{0.100}{2})\right]$$

As $t_{29}(0.050) = 1.699$ this yields the interval $[0.9331, 0.9485]$. This interval and the 0.90 confidence interval we deduced earlier are approximately the same. Keep in mind that we can only use the confidence interval based on the t distribution, if we are convinced that the average occupation rate per subrun is approximately normally distributed. Especially with a small number of subruns, this condition is extremely important!

4.4 Variance reduction

Using the techniques described above, we can analyze simulation results. Reliable assertions often require long simulation runs, which may turn out cost prohibitive. However, some advanced techniques allow reliable assertions from shorter simulation runs. As we have just seen, there is a linear connection between the size of the confidence interval and the standard deviation of the results measured per subrun. If the standard deviation is doubled, the size of the corresponding confidence interval is also doubled. In order to allow assertions with the same reliability, the number of subruns has to increase fourfold! So by decreasing the standard deviation between the subrun results the reliability will increase or the number of subruns needed will decrease. The standard deviation is the square root of the variance, so decreasing the variance and the standard deviation goes hand in hand. The techniques that focus on decreasing the variance are called *variance reducing techniques* Some well-known techniques are:

Variance reducing techniques

- antithetic variates
- common random numbers
- control variates
- conditioning
- stratified sampling
- importance sampling

We will not explain all of these techniques at length and only summarize the first two.

Antithetic variates

In a simulation experiment random numbers are constantly being used and assigned to random variables. A good random generator will generate numbers that are independent of each other. It is, however, not necessary to generate a new set of random numbers for each subrun. If $r_1, r_2, \ldots, r_n$ are random numbers, so are $(1 - r_1), (1 - r_2), \ldots, (1 - r_n)$! These numbers are called antithetic. If we generate new random numbers for each odd subrun and we use antithetic random numbers for each even subrun, we only need half of the random numbers (if the total number of subruns is even). There is another bonus. The results for subrun $2k - 1$ and subrun $2k$ will probably be negatively correlated. If e.g. subrun $2k - 1$ is characterized by frequent arrivals (e.g. caused by sampling small random numbers), the arrivals in subrun $2k$ will be infrequent since antithetic, thus large numbers will be sampled. If x_{2k-1} and x_{2k} represent the results of two consecutive subruns, in all probability $\mathrm{Cov}[x_{2k-1}, x_{2k}]$ will be smaller than zero. This leads to a decrease in the variance of the mean of x_{2k-1} and x_{2k}, as:

$$\mathrm{Var}[\frac{X+Y}{2}] = \frac{1}{4}(\mathrm{Var}[X] + \mathrm{Var}[Y] + 2\mathrm{Cov}[X, Y])$$

The total sample variance will then also decrease, narrowing down the confidence interval.

Common random numbers

If one wants to compare two alternatives, it is intuitively obvious that the circumstances should be as similar as possible. This means that the samples taken in the simulation runs of either alternative should correspond maximally. When simulating different arrangements of the post office, the alternatives may use the same random numbers for interim arrival times and service times. In this way the variance of the difference between both alternatives may be substantially reduced.

There are various advanced techniques for increasing the information obtained through simulation. These techniques have to be used with great care. More details can be found in [2, 12, 13].

4.5 Sensitivity analysis

Sensitivity

Given a certain model, one can give well-founded estimates for the expected waiting times, occupation rates, fault frequencies etc., by using subruns and calculating confidence intervals. Since these results are based on a specific situation, it is unclear how *sensitive* they are. If we find an estimated average waiting time of 10 minutes for an arrival

process with an average interarrival time of 5 minutes, what would the average waiting time be if the interarrival time is not 5 but 6 minutes? In general a model has a number of *parameters*; adjustable quantities, like average interarrival time, average service time and average response time. For a simulation experiment each of these quantities is given a certain value. This value is often estimated, as the exact value is unknown. Also the probability distribution chosen will only approximate the true distribution of values. It is therefore of the utmost importance to know how sensitive the results are to variations in the model parameters.

Model parameters

A *sensitivity analysis* is carried out to assess dependencies between the model parameters and the results. To test the sensitivity, a number of experiments are conducted with slight variations in parameter settings. These experiments indicate the extent to which slight variations can influence the final result. Adjusting the setting of a certain parameter will often only mildly influence on the final result. Sometimes, however, a slight adjustment of a parameter will lead to completely different results. A resource with a high occupation rate will be more sensitive to fluctuations in its arrival process than a resource with a lower occupation rate. We also use the term *robustness*. A model is robust if slight deviations in parameter settings barely influence the final result.

Sensitivity analysis

5 Pitfalls

As already indicated in the introduction, simulation is often used to support critical strategic decisions, where errors are very expensive. However, errors are manifold and often easily introduced when conducting a simulation study, so one has to be on the alert constantly. We will look at the phases in the life-cycle of a simulation study and identify the dangers in each specific phase. Figure 14 shows the phases of a simulation study and the possible sources of errors.

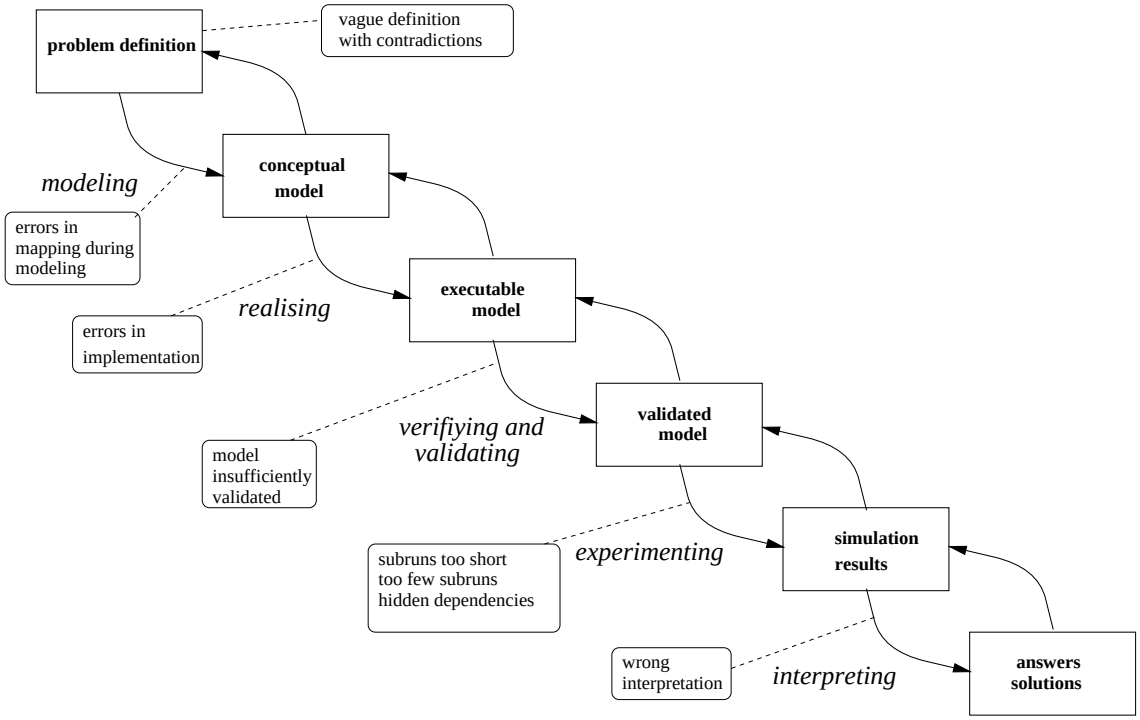

Figure 14: The dangers per phase in a simulation study.

Possible error sources

The problem definition can be inconsistent (contradictory) or incomplete (vague). A conceptual model is developed by a systems analyst and not by the user himself, so various mapping errors can creep in during modeling. Implementing the conceptual model in a simulation language can also introduce errors. If validation is performed by the wrong persons or without the proper care, errors made earlier are not eliminated. Therefore, preferably the model should be validated by the user. During experimentation, errors can arise from too short or too few subruns or from hidden dependencies between the subruns. Also, the initial run can be too short. Errors during experimentation lead to incorrect results. Even if all the previous traps have been avoided, things can still go wrong during interpretation, if faulty conclusions are drawn from the results gathered.

We list ten typical errors (pitfalls) frequently made. Anyone involved in a simulation study should be aware of them and avoid them and their likes.

Error 1: One-sided problem definition

A simulation study gets off on the wrong foot if the problem definition is drawn up exclusively by either the user or the systems analyst. The user may possess extensive knowledge of the problem area, but lacks the experience needed for defining his problem. The systems analyst on the other hand, fully knows the elements which should be present in a problem definition, but lacks the background of the specific problem. The systems analyst is also aware of the possibilities and impossibilities of simulation. The user on the other hand, generally knowing little about simulation, is barely informed on this issue. Therefore, for a simulation study to be successful, it is important that both parties closely cooperate in setting up the problem definition. The problem definition serves as a 'contract' between the user and the builder of the model. A *rule of thumb* for this situation is:

Rule of thumb

> "Do not start a simulation study until it is clear to both user(s) and analyst(s) which questions need to be answered!"

Error 2: Choice of a wrong level of detail

In making a simulation model, one chooses a certain level of detail. In a simulation model for a manufacturing department, a machine may be modeled as an object with serving time as its only parameter. Alternatively, it can be modeled in detail, taking into account aspects such as set-up times, faults, tool-loading, maintenance intervals etc. Many simulation studies are aborted because a wrong level of detail was chosen initially. Too much detail causes the model to become unnecessarily complex and introduces extra parameters that need to be assessed (with all the risks involved). A lack of adequate detail can lead to a simulation model that leaves the essential questions of the problem definition unanswered. The right level of detail is chosen if:

Characteristics of a good level of detail

(1) information is present that allows experiments with the model,

(2) the important questions from the problem definition are addressed by the model and

(3) the complexity of the model is still manageable for all parties concerned.

If it is impossible to choose a level of detail that meets this condition, the problem definition will have to be adjusted.

Error 3: Hidden assumptions

During the modeling and the realization of a simulation model, many assumptions must be made. Assumptions are made to fill gaps in an incomplete problem definition or because of a conscious decision to keep the simulation model simple. Often these assumptions are documented poorly if documented at all, which earns them the name 'hidden assumptions'. Hidden assumptions may lead to the rejection of the simulation model (with or without simulation results) during validation or later. Therefore assumptions must be documented and regularly discussed with the user. In this way future surprises are avoided.

Error 4: Validation by the wrong people

Sometimes, due to time pressure or indifference of the user, the simulation model is only validated by its maker(s). Discrepancies between the model and the ideas of the user may thus be discovered too late, if at all. Therefore, the user should be involved in the validation of the simulation model before any experiments are conducted.

Error 5: Forcing the model to fit

Frequently, in the validation phase, the results of the simulation model do not match the observed or recorded actual data. One is then tempted to make the model 'fit' by changing certain parameter values. One fiddles around with the parameter settings until a match is found. This, however, is very dangerous, since this match with reality is most likely caused by sheer luck and not by a model that adequately reflects reality. Parameters should be adjusted only after having understood why the model deviates from reality. This prevents the conscious or unconscious obscuring of errors in the model.

Error 6: Underexposure of the sensitivity of the model

Certain model parameters (e.g. the intensity of the arrival process) are often set at one specific value. This chosen setting should be justified statistically. However, even if this is the case, small variations in the arrival process can make all assumptions about it invalid. Therefore, the sensitivity of the model to minor adjustments of the parameters should be seriously accounted for.

Error 7: No subruns

Some people say: "A sufficiently long simulation yields correct results!"

They execute a simulation run for a night or weekend and then blindly trust e.g. the mean waiting time measured. This is a very risky practice, as it disallows any assertions about the reliability of the result found. Others derive a confidence interval from the mean variance measured. This is also wrong because the mean variance of the waiting time measured is not connected to the reliability of the estimated mean waiting time, as there exist dependencies between the waiting times of consecutive customers. The only way to derive independent measurements is by a division into subruns!

Error 8: Careless presentation of the results

Interpreting the results of a simulation study may require complex statistical analysis. This is often a source of errors. Translating the results from statistics into language a user can understand, can be very tricky indeed. In Darrel Huff's book "How to lie with statistics" ([4]), there are numerous examples of sloppy and misleading presentations. As an example, suppose the final report of a simulation study contains the following conclusion "Waiting times will be reduced by 10 percent". This conclusion is very incomplete, as it contains no reference whatsoever to its reliability. It is good practice to give a confidence interval. The same conclusion suggests that waiting times will be reduced by 10 percent for each customer. This, however, may not be the case. The average waiting time may be reduced by 10 percent while it increases for certain customers and is reduced somewhat more for others.

Error 9: Dangers of animation

Modern simulation tools allow a pretty presentation of results and possess animation facilities. This improves communication with the user. However, there is a large inherent danger in animation. As animation only shows the tangible aspects of the simulation model, the user may develop an unfounded faith in the model. The choice of parameters or decision making rules deeply influence the simulation results, yet are barely visible in an animation. Also, pretty pictures do not replace a sound statistical analysis.

Error 10: Unneccesary use of simulation

Simulation is a flexible and varied analysis tool. Some people therefore are inclined to use it regardless of the circumstances. Often, however, a simple mathematical model (e.g. a queuing model) or a simple spreadsheet calculation is sufficient. In such cases simulation is 'overkill'. It should only be used if and when the situation requires it. Simulation is a means and not a goal!

6 Closing remarks

In spite of the many caveats in this handbook, simulation is a varied, flexible and dependable analysis tool if applied properly.

We recommend the following books for further reading. A classic book on simulation is the book by Naylor et al. [11]. Other standard works about simulation are: Bratley, Fox and Schrage [2], Law and Kelton [9], Ross [13] and Shannon [14]. In particular the books of Ross ([13]) and Bratley, Fox and Schrage ([2]) are recommended for further study. Books on simulation in Dutch are Kleijnen and Groenendaal [6] and Kerbosch and Sierenberg [5].

References

[1] F. Baskett, K.M. Chandy, R.R. Muntz, and F.G. Palacios. Open, Closed and Mixed Networks of Queues with Different Classes of Customers. *Journal of the Association of Computing Machinery*, 22(2):248–260, April 1975.

[2] P. Bratley, B.L. Fox, and L.E. Schrage. *A guide to simulation.* Springer-Verlag, Berlin, 1983.

[3] D. Gross and C.M. Harris. *Fundamentals of queueing theory.* Wiley, London, 1985.

[4] D. Huff. *How to lie with statistics.* Penguin Books, New York, 1954.

[5] J.A.G.M. Kerbosch and R.W. Sierenberg. *Discrete simulatie met behulp van ALGOL, FORTRAN, PL/1.* Samsom, Alphen aan den Rijn, 1973.

[6] J. Kleijnen and W. van Groenendaal. *Simulatie: technieken en toepassingen.* Academic Service, Schoonhoven, 1988.

[7] J. Kleijnen and W. van Groenendaal. *Simulation: a statistical perspective.* John Wiley and Sons, New York, 1992.

[8] L. Kleinrock. *Queueing systems, Vol. 1:Theory.* Wiley-Interscience, London, 1975.

[9] A.M. Law and D.W. Kelton. *Simulation modeling and analysis.* McGraw-Hill, New York, 1982.

[10] M. Ajmone Marsan, G. Balbo, and G. Conte. *Performance Models of Multiprocessor Systems.* The MIT Press, Cambridge, 1986.

[11] T.H. Naylor, J.L. Balintfy, D.S. Burdick, and Kong Chu. *Computer simulation techniques.* Wiley, New York, 1966.

[12] M. Pidd. *Computer modelling for discrete simulation.* John Wiley and Sons, New York, 1989.

[13] S.M. Ross. *A course in simulation.* Macmillan, New York, 1990.

[14] R.E. Shannon. *Systems simulation: the art and science.* Prentice-Hall, Englewood Cliffs, 1975.

A Elementary properties

In this appendix we treat some elementary properties of random variables. X and Y are random variables with respectively expectation μ_X and variance σ_X^2, and expectation μ_Y and variance σ_Y^2
Concerning addition and multiplication of variates, the following universal properties apply:

$$\begin{aligned} \mathbb{E}[X+Y] &= \mu_X + \mu_Y \\ \mathbb{E}[aX+b] &= a\mu_X + b \end{aligned}$$

The variance of a random variable X ($\mathrm{Var}[X] = \sigma_X^2$) can be expressed in terms of the expectation of X and X^2.

$$\mathrm{Var}[X] = \mathbb{E}[(X-\mu_X)^2] = \mathbb{E}[X^2] - \mu_X^2$$

For the variance the following property is important:

$$\mathrm{Var}[aX+b] = a^2\sigma_X$$

The covariance of X and Y ($\mathrm{Cov}[X,Y]$) is also denoted as σ_{XY}^2.

$$\mathrm{Cov}[X,Y] = \mathbb{E}[(X-\mu_X)(Y-\mu_Y)] = \mathbb{E}[XY] - \mu_X\mu_Y$$

If X and Y are independent, then $C[X,Y] = 0$.
There is also a relation between variance and covariance:

$$\mathrm{Var}[X+Y] = \sigma_X^2 + \sigma_Y^2 + 2\mathrm{Cov}[X,Y]$$

So if X and Y are independent, $\mathrm{Var}[X+Y] = \sigma_X^2 + \sigma_Y^2$.

A.1 Markov's inequality

If a random variable X only takes on non-negative values, then for each $a > 0$:

$$\mathbb{P}[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$$

A.2 Chebyshev's inequality

Given a random variable X with mean μ and variance σ^2, then for each $k > 0$:

$$\mathbb{P}[|X-\mu| \geq k\sigma] \leq \frac{1}{k^2}$$

A.3 Central limit theorem

For a set $X_1, X_2, \ldots, X_n$ of independent uniformly distributed random variables with expectation μ and variance σ^2, the random variable

$$\frac{(X_1 + X_2 + \ldots + X_n) - n\mu}{\sigma\sqrt{n}}$$

converges for $n \to \infty$ to a standard normal distribution.
Thus, the sum of a large number of independent random variables is approximately normally distributed. In interpreting simulation results we can assume (for $n > 10$) that the sum of n independent random variables X_i with expectation μ and variance σ^2, is approximately normally distributed with expectation $n\mu$ and variance $n\sigma^2$. A similar statement holds for the mean of the X_i.

A.4 The extent of the normal distribution

The central limit theorem above shows the importance of the normal distribution for the interpretation of simulation results. Given a normal distribution with parameters μ (expectation) and σ (standard deviation), the probability of a sample lying between $\mu-\sigma$ and $\mu+\sigma$ is approximately 0.683. The following table shows for a number of intervals surrounding μ the probability of a draw from a normal distribution with parameters μ and σ in this interval.

interval	probability
$[\mu - \frac{\sigma}{2}, \mu + \frac{\sigma}{2}]$	0.383
$[\mu - \sigma, \mu + \sigma]$	0.683
$[\mu - 2\sigma, \mu + 2\sigma]$	0.954
$[\mu - 3\sigma, \mu + 3\sigma]$	0.997

So, the probability that a certain draw is between $\mu - 3\sigma$ and $\mu + 3\sigma$ is approximately 0.997.

B Summary random distributions

B.1 Discrete random distributions

distribution	**domain**	$\mathbb{P}[X=k]$	$\mathbb{E}[X]$	$\mathrm{Var}[X]$
Bernoulli $0 \leq p \leq 1$	$k \in \{0,1\}$	$\begin{cases} 1-p & k=0 \\ p & k=1 \end{cases}$	p	$p(1-p)$
homogeneous $a < b$	$k \in \{a, \ldots, b\}$	$\frac{1}{(b-a)+1}$	$\frac{a+b}{2}$	$\frac{(b-a)((b-a)+2)}{12}$
binomial $0 \leq p \leq 1$ $n \in \{1,2,\ldots\}$	$k \in \{0,1,\ldots,n\}$	$\binom{n}{k} p^k (1-p)^{n-k}$	np	$np(1-p)$
geometric $0 \leq p \leq 1$	$k \in \{1,2,\ldots\}$	$(1-p)^{k-1}p$	$\frac{1}{p}$	$\frac{1-p}{p^2}$
Poisson $\lambda > 0$	$k \in \{0,1,\ldots\}$	$\frac{\lambda^k}{k!}e^{-\lambda}$	λ	λ

B.2 Continuous random distributions

distribution	**domain**	$f_X(x)$	$\mathbb{E}[X]$	$\mathrm{Var}[X]$
uniform $a < b$	$a \leq x \leq b$	$\frac{1}{b-a}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
exponential $\lambda > 0$	$x \geq 0$	$\lambda e^{-\lambda x}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
normal $\mu \in \mathbb{R}$ $\sigma > 0$	$x \in \mathbb{R}$	$\frac{1}{\sqrt{2\pi\sigma^2}}\ e^{\frac{-(x-\mu)^2}{2\sigma^2}}$	μ	σ^2
gamma $r, \lambda > 0$	$x > 0$	$\frac{\lambda(\lambda x)^{r-1}\ e^{-\lambda x}}{\Gamma(r)}$	$\frac{r}{\lambda}$	$\frac{r}{\lambda^2}$
Erlang $\lambda > 0$ $r \in \{1,2,\ldots\}$	$x > 0$	$\frac{\lambda(\lambda x)^{r-1}\ e^{-\lambda x}}{(r-1)!}$	$\frac{r}{\lambda}$	$\frac{r}{\lambda^2}$
χ^2 $v \in \{1,2,\ldots\}$	$x > 0$	see gamma $r = \frac{v}{2}$ and $\lambda = \frac{1}{2}$	v	$2v$
beta $a < b$ $r, s > 0$	$a \leq x \leq b$	$\frac{1}{b-a}\ \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)}$ $\left(\frac{x-a}{b-a}\right)^{r-1}\left(\frac{b-x}{b-a}\right)^{s-1}$	$a + (b-a)\frac{r}{r+s}$	$\frac{rs(b-a)^2}{(r+s)^2(r+s+1)}$

C Queuing models

Queuing models

So-called analytical models can be analyzed directly without simulation. A well-known class of such models are the so-called *queuing models*. Here, we only treat single queue models. Important results have been derived for networks of queues (see e.g. Baskett et al. [1] and Marsan et al. [10]).

$A/B/c$-notation

A queue is often characterized by $A/B/c$ where A refers to the arrival process, B to the distribution of the serving times and c to the number of parallel identical servers. The letter M is used to indicate a negative exponential distribution. The letter E_r refers to an Erlang distribution. G denotes an arbitrary distribution.
One of the simplest queuing models is the $M/M/1$ queue, i.e. a queue with a Poisson arrival process (the interarrivals are negative exponentially distributed), negative exponentially distributed serving times and only 1 server (at most one customer is served at a time).

Little's formula

Before presenting some results, we will state *Little's formula*, which applies to each queue in a stable state without dependencies between the arrival process and the serving process. The formula is:

$$L = \lambda S$$

where L is the mean number of customers in the system, λ the mean number of customers arriving per time unit and S the mean system time (i.e. the time that customers stay within the system on average, so the sum of the waiting time and the serving time).

$M/M/1$ queue

The $M/M/1$ queue is specified by two parameters λ and μ. For the arrival process the negative exponential distribution with parameter λ is used. The mean interarrival time is thus $\frac{1}{\lambda}$. For the serving process a negative exponential distribution with parameter μ is used. The mean serving time is thus $\frac{1}{\mu}$. The occupation rate ρ is:

$$\rho = \frac{\lambda}{\mu}$$

We can represent the state of the queue by the integer k that represents the total number of customers in the system. The probability that the queue is in state k at a given moment, is denoted as p_k:

$$p_k = (1-\rho)\rho^k$$

These probabilities are also called the *steady-state probabilities*. The mean number of customers in the system is L:

$$L = \frac{\rho}{1-\rho}$$

The mean systems time is S:

$$S = \frac{1}{(1-\rho)\mu}$$

The mean waiting time W is the difference between the systems time and the serving time:

$$W = \frac{\rho}{(1-\rho)\mu}$$

$M/E_r/1$ queue

The $M/E_r/1$ queue is specified by three parameters λ, r and μ. The serving time now is distributed Erlang with parameters λ and r. For the occupation rate ρ, the mean number of customers L, the mean system time S and the mean waiting time W, we find the following values:

$$\begin{aligned}
\rho &= \frac{r\lambda}{\mu} \\
L &= \frac{\lambda\rho(r+1)}{2\mu(1-\rho)} + \rho \\
S &= \frac{\rho(r+1)}{2\mu(1-\rho)} + \frac{r}{\mu} \\
W &= \frac{\rho(r+1)}{2\mu(1-\rho)} + \frac{r-1}{\mu}
\end{aligned}$$

$M/G/1$ queue

Less is known about the $M/G/1$ queue Once again, the arrival process is specified by the parameter λ. The mean serving time is $\frac{1}{\mu}$ and the variance of the serving time is σ^2. The variance of the serving time C is defined as follows: $C = \sigma\mu$. For the occupation rate ρ, the mean number of customers L, the mean system time S and the mean waiting time W we find the following values:

$$\begin{aligned}
\rho &= \frac{\lambda}{\mu} \\
L &= \rho + \frac{\rho^2}{2(1-\rho)}(1+C^2) \\
S &= \frac{1}{\mu} + \frac{\rho}{2\mu(1-\rho)}(1+C^2) \\
W &= \frac{\rho}{2\mu(1-\rho)}(1+C^2)
\end{aligned}$$

The first formula is also known as Pollaczek-Khinchin's formula.

$M/G/\infty$ queue

The $M/G/\infty$ queue is specified by two parameters λ and μ. The variance

of the distribution G is not relevant. Because there are always a sufficient number of free servers in the $M/G/\infty$ queue, the occupation rate cannot be defined (actually, it is 0). We now equate ρ with the amount of work arriving per time unit: $\rho = \frac{\lambda}{\mu}$. The steady-state probabilities p_k now are:

$$p_k = \frac{\rho^k}{k!} e^{-\rho}$$

For the mean number of customers L, the mean system time S and the mean waiting time W, we find the following values:

$$\begin{aligned} L &= \rho \\ S &= \frac{1}{\mu} \\ W &= 0 \end{aligned}$$

For more information on this subject we refer you to Kleinrock [8] and Gross and Harris [3].