Wednesday 18th July, 2007

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Sequential Analysis of Quantiles and Probability Distributions by Replicated Simulations

A thesis submitted in partial fulfilment
of the requirements for the Degree of
Doctor of Philosophy in Computer Science
in the University of Canterbury by
Mirko Eickhoff

Examining Committee and Supervisors

Associate Professor Mark Bebbington (Examiner)
(Massey University, Information Sciences & Technology)

Professor Peter W. Glynn (Examiner)
(Stanford University, Management Science & Engineering)

Professor Krzysztof Pawlikowski (Examiner & Supervisor)
(University of Canterbury, Computer Science & Software Engineering)

Associate Professor Don McNickle (Supervisor)
(University of Canterbury, Management)

To Nadine

Abstract

Discrete event simulation is well known to be a powerful approach to investigate behaviour of complex dynamic stochastic systems, especially when the system is analytically not tractable. The estimation of mean values has traditionally been the main goal of simulation output analysis, even though it provides limited information about the analysed system's performance. Because of its complexity, quantile analysis is not as frequently applied, despite its ability to provide much deeper insights into the system of interest. A set of quantiles can be used to approximate a cumulative distribution function, providing fuller information about a given performance characteristic of the simulated system.

This thesis employs the distributed computing power of multiple computers by proposing new methods for sequential and automated analysis of quantile-based performance measures of such dynamic systems. These new methods estimate steady state quantiles based on replicating simulations on clusters of workstations as simulation engines. A general contribution to the problem of the length of the initial transient is made by considering steady state in terms of the underlying probability distribution. Our research focuses on sequential and automated methods to guarantee a satisfactory level of confidence of the final results. The correctness of the proposed methods has been exhaustively studied by means of sequential coverage analysis. Quantile estimates are used to investigate underlying probability distributions. We demonstrate that synchronous replications greatly assist this kind of analysis.

Contents

List of Figures

List of Tables

Notation

CDF	cumulative distribution function
PDF	probability density function
IID	independent and identically distributed
X	random variable
$F_X(x)$	CDF of X
$f_X(x)$	PDF of X
$F_X^{-1}(q)$	inverse of $F_X(x)$
x	observation of X
$\{X_i\}_{i=1}^p$	random sample or stochastic process
$\{x_i\}_{i=1}^p$	realisation of $\{X_i\}_{i=1}^p$
$\{Y_i\}_{i=1}^p$	sorted sample of $\{X_i\}_{i=1}^p$, where $Y_1 \leq \ldots \leq Y_p$
$x_{j,i}$	ith observation of jth replication
$\mathrm{E}[X]$	expected value (1st moment)
$\mathrm{Var}[X]$	variance (2nd central moment)
$\mathrm{Cov}[X_0, X_1]$	covariance
$\mathrm{Skew}[X]$	skewness (3rd standardised moment)
$\mathrm{Kurt}[X]$	kurtosis (4th standardised moment)
$\mathrm{U}(x; a, b)$	uniform distribution with bounds a and b
$\mathrm{Exp}(x; m^{-1})$	exponential distribution with mean m
$\mathrm{N}(x; m, v)$	normal distribution with mean m and variance v
Fib_k	kth Fibonacci number

$\mathsf{KS_k}$	k-sample Kolmogorov-Smirnov test statistic
$\mathsf{AD_k}$	k-sample Anderson-Darling test statistic
$\bar{x}$, $\hat{x}$, $\hat{X}$	average, estimate, estimator
$\lfloor x \rfloor$	largest integer equal or smaller than x
$\lceil x \rceil$	smallest integer equal or greater than x
n	simulation horizon
p	number of replications
i	observation index
j	replication index
l, l_F, l_E, l_V	truncation point
r	ratio
q	probability
x_q	position of q-quantile
λ	arrival rate
μ	service rate
ρ	traffic load
$1-\alpha$	confidence level
ϵ	threshold
Δ	interval, difference, halfwidth
$\{\Psi_i\}_{i=1}^{\infty}$	independent Gaussian white noise process
$\{\Upsilon_i^{(k)}\}_{i=1}^{\infty}$	geometrical ARMA(k,k) process
N_i	queue length at arrival (resp. departure) of ith customer
R_i	response time of ith customer

This is a list of commonly used notations and symbols of this thesis. Exceptions of these notations cannot be avoided and are stated in the context of the associated section.

Acknowledgement

I wish to thank my supervisors, Professor Krzysztof Pawlikowski and Associative Professor Don McNickle, for helpful advice, patient guidance, enlightening discussions and valuable feedback on drafts of my thesis. I am grateful for the support of all members of the Simulation Research Team at various stages of my work. Many thanks to all people of the Computer Science & Software Engineering Department for providing best working conditions and computer facilities.

Finally, I wish to thank my family and friends who contributed to this thesis by supportive talks, encouragement, giving me direction as well as distraction and making New Zealand my second home.

This research was supported by a targeted doctoral scholarship granted by the University of Canterbury and by travel grants for participation at international conferences granted by the Computer Science & Software Engineering Department and the conference chair of the MMB 2006.

Chapter 1

Introduction and Motivation

Stochastic discrete event simulation is well known to be a powerful approach to investigate dynamic behaviour of complex systems, especially when the system is analytically not tractable. Nowadays much research work in this area is focused on sequential and automated methods of output analysis to guarantee a satisfactory level of confidence of the final results.

The estimation of mean values has traditionally been the main goal of simulation output analysis. Mean value estimation enables the analyst to answer questions of the kind: What is the average delay of a data packet passing a server? What is the average filling of a storage? What is the average utilisation of a worker at an assembly line. However, in many situations mean value analysis is not sufficient. The estimation of quantiles is known to provide the analyst with a deeper insight into the system's behaviour. Quantile estimation enables the analyst to answer questions like: What is the probability of a file transfer in the Internet being delayed for more than x seconds? What is the probability of overloading a machine with too many jobs? What is the probability of a storage being empty?

The complexity of quantile analysis is higher than the complexity of mean value analysis due to higher complexity of estimators. However, the main problem facing quantile estimation is the same as that of mean value analysis: output

streams from discrete event simulation are autocorrelated and observations are not identically distributed. The ultimate goal of estimating the steady state distribution is, therefore, not straightforward. The use of independent replications performed in parallel within one simulation experiment enables the investigation of effective statistical methods of quantile analysis and offers a new paradigm for studying performance of complex systems.

The aim of this doctoral thesis is to propose new methods for automated and sequential analysis of performance measures of such dynamic systems and to investigate the underlying probability distributions based on quantile analysis. The results are calculated with a certain confidence level given by a sequential and automated approach within the scenario of independent replications. Estimation of quantiles in steady state requires to define the onset of steady state conditions for such analysis. One of the main results of this thesis is a novel technique for determining the length of the initial transient phase by detecting the convergence of output processes to their steady state probability distribution. These novel solutions will be discussed in more detail in the following sections.

1.1 Simulation as a Statistical Experiment

Perhaps the most widely used paradigm of system analysis and optimisation is *stochastic discrete event simulation*. Compared to other paradigms its main advantage is that it can be used for studying analytically intractable systems, as long as procedures describing their behaviour are known. Therefore, the areas of application of simulation is vast, e.g. telecommunication networks, manufacturing systems, logistic networks and many more. Every stochastic simulation is a statistical experiment due to the random property of the simulation inputs. Thus, the simulation results can be considered as estimates of true characteristics of the model.

Credibility of simulation depends mainly on two factors. First, the simulation model must be able to mimic the wanted behaviour. Verification and validation (see [85-LK00]) of models is needed. To simplify the creation of a valid model many simulation software packages are specialised to certain application areas. For example [18-BTD06] and [83-KCC05] discuss credibility of simulation for certain kinds of networks and show common pitfalls. However, we focus on the second factor of credibility and follow the principles of [101-PJL02]. In this article a survey of recent publications showed that surprisingly, up to three quarters of papers reporting simulation-based results did not acknowledge the random nature of output data generated by stochastic simulation. The output of simulations must be processed in a statistically valid way to be useful to the decision maker. Estimates of true characteristics of a simulation model must be given with a controlled statistical error. Bias of estimators should be controlled, or even eliminated. Unfortunately, much commercial software focuses on secondary properties of simulators only, for example 3D-animation. Neglecting the need to produce final results with small accurately estimated statistical errors can lead to very inaccurate results and to a loss of simulation's credibility in general.

For this reason we use *sequential and automated simulation analysis* and follow the paradigm of [99-Paw90]. For an introduction to sequential procedures see [85-LK00]. In sequential analysis the simulation runs are guided by the process of analysis. Their length is sequentially increased until the statistical error of the measure of interest is small enough. This is defined by a sequential stopping criteria. This implies that in sequential analysis the simulation run length is not fixed before the simulation experiment. Sequential analysis is the only way to obtain estimates with controlled statistical error. Confidence intervals of estimates should be calculated on a reasonably high confidence level to provide meaningful results. Additionally, the width of confidence intervals should be small. This means that e.g. the relative width is smaller than a predefined small threshold. This can be

guaranteed only if output analysis is performed during to the simulation run. The simulation experiment is continued until the predefined threshold is reached. In automated analysis the interaction with the user should be minimised. Setting parameters like confidence levels or acceptably relative errors should not be an issue and could be even avoided by choosing standard values. Additionally, the user should not be obliged to set parameters which require deep understanding of simulation models, like e.g. correlation structures or steady state characteristics. Our aim is to provide sequential and automated methods for output analysis of stochastic discrete event simulation. The software, which is developed for the purpose of this thesis, is designed to be part of a universal simulation controller, like *Akaroa2* ([47-EPM99]), that supports data collection, sequential analysis and stopping simulation when results become satisfactorily accurate. This kind of simulation output data analysis is online, because data is analytically processed as soon as it is observed.

1.2 Focus of Output Analysis

In analysis of simulation output data many different performance measures could be of interest, such as the different central moments, gradients, probabilities, rare events or quantiles. Mean value analysis is the most common approach in simulation. Mean values describe average system behaviour. Statistical errors in mean value analysis can be done by estimating confidence intervals on basis of the variance of the mean. This includes batch means or spectral analysis, which have been discussed in [99-Paw90]. Variance analysis can also be done to determine the variance of the underlying output process itself. Gradient estimation (see e.g. [57-Gly90]) is done to find optimal settings of a model. Here, not only a given model is of interest but the optimisation of this model. To analyse the behaviour of a system in extreme situations, rare event simulation (see e.g [120-Sha95] or

[132-VAVA94]) is applied. This simulation approach targets at the estimation of measures which are based on events with very low probability. All in all we can see that methods are available to estimate measures of average and extreme system behaviour. However, a deeper insight into the systems behaviour can be given by estimating the full spectrum of possible values of a measure and their probability of occurrence, i.e. by estimating the probability distribution of the measure of interest.

An impression of a probability distribution can be obtained by a set of sufficiently many, suitably spaced quantiles. Thus, *quantile estimation* of the simulation output process is an important task and will be the main focus in later chapters. We will review basic mathematics of quantile estimation in Section 2.2 followed by a survey in Section 2.4 of the current situation of quantile estimation in simulation. We will derive quantile estimation methods for their time-dependent, as well as for their steady state behaviour. The application areas of quantile estimation are as vast as the application areas of simulation itself. Inventory systems, queueing systems, computer systems, real-time control applications, financial industry, Internet and many more are explicitly mentioned in literature as areas of applications (see e.g. [74-Ige76], [76-JC85], [48-FMG$^+$01] or [77-JFX03]). An application in the area of peer-to-peer file sharing systems is discussed in Section 4.4.4 and shows the importance of the new methods in a practical example. Throughout all chapters, the use of new derived methods will be demonstrated on examples. In addition, important statistical properties will be proven analytically.

Our focus are methods for automated simulation. We show that the proposed methods of this thesis are robust and applicable for all kinds of output data in general. However, in the description of these methods we will rely on assumptions, which are done to show statistical properties. If these assumptions are violated, the methods do not necessarily fail; it is more likely that the results are not as good as one would expect for valid assumptions. *Limits* of the new methods are

given by the assumed properties of underlying probability distributions. Namely, we assume continuous distribution functions so that the usual definition of quantiles (see Equation (2.12)) is valid. This will effect methods of Chapter 4 and Chapter 6. Assuming continuous distribution functions is also beneficial when comparing random samples: the usual statistics of homogeneity tests (Chapter 5) are applicable. Furthermore, all methods are tested for well behaved distributions only, since we assume that lower moments of the underlying distribution should be finite. If lower moments are not finite, estimating the probability distribution is questionable anyway.

1.3 Scientific Contribution

The following list is a summary of the main scientific contributions of this thesis.

- Analysis of the time evolution of a stochastic output process based on a set of quantile estimates: Chapter 4, which is published in [40-EMP05a], [42-EMP06]. Application in analysis of time dependent file popularity in Peer-to-Peer networks, which is published in [17-BEPS07].

- Subdividing the output process in a initial transient and steady state phase in terms of a stable probability distribution: Chapter 5, which is published in [41-EMP05b], [38-Eic06] and [43-EMP07a].

- Approximation of the steady state probability distribution based on a set of quantile estimates: Chapter 6, which is published in [38-Eic06] and [44-EMP07b].

This doctoral thesis is supported by the following list of refereed conference contributions and journal articles.

2005	[40-EMP05a]	Depiction of Transient Performance Measures using Quantile Estimation (ECMS)
	[41-EMP05b]	Efficient Truncation Point Estimation for Arbitrary Performance Measures (ISC)
2006	[42-EMP06]	Analysis of the Time Evolution of Quantiles in Simulation (IJSSST)
	[38-Eic06]	Steady State Quantile Estimation (MMB)
2007	[17-BEPS07]	Modeling File Popularity in Peer-to-Peer File Sharing Systems (ASMATA)
	[43-EMP07a]	A Method for Detecting the Initial Transient in Steady State Simulation of Arbitrary Performance Measures (VALUETOOLS)
	[44-EMP07b]	Using Parallel Replications for Sequential Estimation of Multiple Steady State Quantiles (VALUETOOLS)

Table 1.1: List of refereed conference contributions and journal articles.

1.4 Structure of Thesis

The *outline* of this thesis is as follows. In the next chapter we will give an overview of the different performance measures of simulation. We will focus especially on quantiles of the simulation output process. This is followed by the description of a simulation scenario where independent replications are performed in parallel. This will provide a speed up in data collection on the one hand. On the other hand, we will explore the use of replications to introduce new estimators. In Chapter 4 we will calculate a set of quantiles and estimate their time dependent evolution. The aim is to provide information about the dynamics of a model. A different point of view is considered in Chapter 5. Here, we describe methods that distinguish between the transient and the steady state behaviour of the simulation output process. Homogeneity test will be very important for this task. They enable us to find steady state, so that identically distributed output data can be assumed. These methods are important for subsequent analysis of steady state measures. In Chapter 6 we discuss quantile estimation during steady state. A set of quantiles is calculated on basis of multiple parallel replications. This results in an estimate of the steady state probability distribution.

Chapter 2

Performance Measures in Simulation

In this section we comment on output data analysis in simulation. Because there are many different kinds of performance measure we give an overview and introduce necessary definitions in Section 2.1. We describe and distinguish probabilistic characteristics of a model, estimators and the output processes themselves. Quantiles are special characteristics, which will be discussed in general terms in Section 2.2. The case of a sample with independent and identically distributed observations is discussed. Point and interval estimators for quantiles are derived on basis of order statistics. In Section 2.3 a selection routine for a set of quantiles is discussed. The special role of quantile estimation in simulation output data is discussed in Section 2.4. Here, the data cannot be assumed to be independent and identically distributed. We survey simulation literature regarding this topic. In general we have to distinguish between the estimation of one, or several quantiles as well as between finite horizon or steady state simulation. Two selected approaches are reviewed at the end of this section.

2.1 Output Process and Estimation

In stochastic simulation the analyst has to deal with a lot of different performance measures which are all governed by certain probability distributions. All these measures are dependent on each other, but there are different analytical layers.

We consider simulation models of systems whose states are probabilistic. So are the observations which are taken from such simulations. In a simulation experiment the stream of observations forms an output process that commonly shows random and time dependent behaviour. The estimates of characteristics of this output process are probabilistic, too. This will be discussed in this section.

For more details and explanations of the terminology which is used in the following sections, we refer to standard simulation literature such as [51-Fis01], [85-LK00], [23-CL99], [10-Ban98], [11-BCN96], [75-Jai91] and [21-BFS87] or textbooks such as [121-Sha75], [49-Fis73], [95-Mih72], [46-ES70], [65-Gor69] and [130-Toc63].

2.1.1 Characteristics of Models

In general, a *model* is an abstract representation of a system for the purpose of studying this system. The model describes a selection of entities, attributes, activities and events of this system, which are important for the specified aim of the study. In stochastic discrete event simulation a model can be implemented in any kind of computer programming language which is appropriate.

The model is fully described by all its *state variables*. For example such quantities as W =“number of waiting customers in a queue”, L =“water level of a pool”, C =“condition of a teller” or T =“activity of a printer” are examples for states and can be described by random variables, which will be denoted with uppercase letters. As we can see already of this short list of examples, there are many different types of state variables:

- W is a *discrete* random variable greater than zero;
- L is a *continuous* random variable greater than zero;
- C is a *binary* random variable, e.g. busy or idle;
- T is a *discrete* random variable, e.g. "printing", "receiving data" and "waiting".

Realisations of random variables will be denoted by corresponding lowercase letters, e.g. $W = \{w_1, w_2, \ldots\}$ or $T = \{t_1, t_2, \ldots\}$.

In discrete event simulation a model changes its states instantaneously whenever some event occurs. For the purpose of analysis *observations* are collected during the simulation. This may happen at the occurrence of an event. An observation can be the value of a state variable as well as a measure of its property. Thus, there are many different kinds of observations possible, as they can inform about, for example, response time, waiting time, service time, turnaround time, interarrival time, queue length, filling of a storage, utilisation, arrival rate, exit rate, etc. A simulation is able to provide the analyst with an unbounded amount of observations. The problem is how to use these data in a decision process associated with performance evaluation of modern computer dynamic systems. These observations have to be statistically processed to enable precise evaluation of the systems concerned.

2.1.2 Characteristics of Output Processes

As we have mentioned in the previous section, a simulation can report many different kinds of observations. Of course, different kinds of observations should not be mixed with each other, but they should be analysed separately. This is why we can focus on only one kind of observation per simulation in the following discussions, without loss of generality. However, in this way correlations between

different measures might be overlooked. On the one hand correlations between different performance measure might provide information about the system of analysis. On the other hand this kind of correlation is an additional difficulty in output analysis, as discussed in [85-LK00] combined confidence intervals have to be calculated. This might be difficult if many different performance measures are analysed and is a reason for looking at single performance measures in isolation, independently one from the other.

Any kind of observation can be described by a random variable X_i with observation index i. Instead of associating an event with its time of occurrence, an event can be associated with its number of occurrences so far: the observation index i. In the case of time-continuous measures, such as queue length, instead of recording data continuously in time, measurements should be done at specific time instances, characteristics for a given process. For example, queue length could be measured at just after arrival of a new customer to a given queue, or just after departure of a customer from a given queue. All observations from a simulation are given by the sequence $\{X_i\}_{i=1}^{n}$ and determine the output process, where X_i is a component of this process. The output process, as a sequence of random variables, forms a stochastic process. In general, however, the components of the output process of a simulation is neither independent nor identically distributed. Because $i \in \mathbb{N}$ the sequence $\{X_i\}_{i=1}^{n}$ is a discrete time stochastic process that can be continuous or discrete valued. $n \in \mathbb{N}$ can be a fixed value, e.g. if a specified number of observations is collected. In contrast to this, note that in following chapters n is sometimes used to describe a temporary simulation horizon. This will be mentioned explicitly or is obvious in the current context. In general, the simulation horizon n is not bounded. To mention this explicitly we will often describe the output process as an infinite sequence $\{X_i\}_{i=1}^{\infty}$.

The cumulative distribution function (CDF) of X_i is given by

$$F_{X_i}(x) = \Pr\left[X_i \leq x\right], \tag{2.1}$$

which is the marginal (or first order) distribution of the process $\{X_i\}_{i=1}^{\infty}$. Distributions of higher order are given by the convolution of marginal distributions at different i. In this case, the values i and x of Equation (2.1) are replaced by vectors; for details see e.g. [131-Tri02]. Possibly interesting characteristics of X_i are for example

- the expected value $\mathrm{E}[X_i]$ (1st moment),
- the variance $\mathrm{Var}[X_i]$ (2nd central moment),
- the skewness $\mathrm{Skew}[X_i]$ (3rd standardised moment),
- the kurtosis $\mathrm{Kurt}[X_i]$ (4th standardised moment),
- as well as for example the mode,
- the median, quartiles, deciles, percentiles, or
- other quantiles of $F_{X_i}(x)$.

By far the majority of simulation literature focuses on the estimation of the expected value $\mathrm{E}[X_i]$. Using estimators of mean values, the results of the simulation can answer questions about the average system state, such as: How many customers are there on average in the queue? However, the estimation of other characteristics could be desirable for the analyst. Especially the estimation of quantiles can additionally answer questions like: What is the probability of more than k customers in the queue? Questions of this kind are often of more interest to the decision-maker. The complexity of quantile estimation is higher than the complexity of mean value estimation, but the estimation of quantiles can give full insight into the system of interest. This is true especially if several quantiles are estimated. A set of several quantiles can be used to approximate $F_{X_i}(x)$.

2.1.3 Characteristics of Estimators

Let $\hat{\Theta}$ be an estimator for unknown Θ based on a collection of observations, where for example $\Theta = \mathrm{E}[X_\infty]$ or $\Theta = F^{-1}_{X_\infty}(0.95)$. The estimator $\hat{\Theta}$ is governed by the CDF $F_{\hat{\Theta}}(x)$. Here, characteristics of $\hat{\Theta}$ are important to evaluate the performance and quality of the estimator.

$\mathrm{E}\left[\hat{\Theta}\right]$ is important for the definition of the bias:

$$\mathrm{B}\left(\hat{\Theta}\right) = \mathrm{E}\left[\hat{\Theta}\right] - \Theta. \tag{2.2}$$

$\hat{\Theta}$ is an unbiased estimate only if $\mathrm{B}\left(\hat{\Theta}\right) = 0$. Whereas $\mathrm{Var}\left[\hat{\Theta}\right]$ is important for the definition of the mean squared error:

$$\mathrm{MSE}\left(\hat{\Theta}\right) = \mathrm{E}\left[\left(\hat{\Theta} - \Theta\right)^2\right] = \mathrm{Var}\left[\hat{\Theta}\right] + \left(\mathrm{B}\left(\hat{\Theta}\right)\right)^2. \tag{2.3}$$

Quantiles of $F_{\hat{\Theta}}(x)$ are used to determine a confidence interval for unknown Θ.

2.1.4 Evolution in Time versus Steady State

The primary interest of the analyst are characteristics of the output process $\{X_i\}_{i=1}^{\infty}$. The ultimate goal of this thesis is to develop techniques for describing the output process by its marginal distributions $F_{X_i}(x)$, where $i < \infty$, and especially by $F_{X_\infty}(x)$, if it exists. This is not a trivial task, which is maybe the reason why most methods of simulation output analysis focus on $\mathrm{E}[X_i]$.

We have to distinguish between two cases. For the first case let us assume that the simulation model is initialised at a particular state at the beginning of the simulation experiment. The analyst might be interested in how the simulated process evolves over time, e.g. recovers from an exceptional event. In this case a fixed simulation horizon is appropriate and we are dealing with finite horizon simulation. The analyst may be interested in quantiles of $F_{X_i}(x)$ evolving over time. A method for this task will be described in Chapter 4.

In the second case the analyst is more interested in the long run behaviour of a simulated system. This usually assumes that there is a common distribution $F_X(x)$ for all X_i with $i \geq l$, which is called the steady state behaviour. Observation indexes with $i < l$ are usually discarded to limit the influence of an arbitrary initial state. In this case the time evolution is not of interest because in steady state $F_{X_i}(x)$ is not changing over time. Here, the ultimate goal is to approximate $F_X(x)$. This will be discussed in Chapter 5 and Chapter 6.

2.2 Quantiles in IID samples

In the previous section we pointed out that in general the estimation of quantiles provides a deeper insight into the behaviour of the system of interest than the estimation of mean values. In this section we would like to discuss the basic mathematics in quantile estimation and introduce the necessary denotation. Our discussion of order statistics and quantiles in this section is mainly based on [34-Dav70], [6-AB89] and [29-Con99].

2.2.1 Order Statistics

Let $X_1, X_2, \ldots, X_p$ be a set of independent and identically distributed random variables, i.e.

$$\Pr\left[\forall i : X_i \leq x\right] = \prod_i \Pr\left[X_i \leq x\right] \tag{2.4}$$

and

$$\forall i : F_{X_i}(x) = F_X(x), \tag{2.5}$$

respectively. Then, $F_X(x)$ is the common CDF of all X_i and can be estimated by

$$\hat{F}_X(x) = \frac{1}{p} \sum_{i=1}^{p} \zeta(x - X_i), \tag{2.6}$$

where

$$\zeta(\Delta) = \begin{cases} 1, & \text{if} \quad \Delta \geq 0, \\ 0, & \text{else.} \end{cases} \tag{2.7}$$

$\hat{F}_X(x)$ is called the empirical CDF. The value of $\hat{F}_X(x)$ is determined by counting how many observations of $\{X_i\}_{j=1}^{p}$ are smaller than x. If k values of $\hat{F}_X(x)$ are of interest, the use of Equation (2.6) leads to a time complexity of $\mathrm{O}\,(kp)$, since a linear search has to be done k times in a sample of size p. This is inefficient. In this situation it is advisable to base the estimation on a sorted random sample.

Let $\{Y_i\}_{i=1}^{p}$ be the ordered sequence of $\{X_i\}_{i=1}^{p}$, i.e. $Y_1 \leq Y_2 \leq \ldots \leq Y_p$. Then y_i is called the ith order statistic and Y_i is the associated random variable. Because $Y_i \leq Y_{i+1}$, order statistic are dependent and not identically distributed. The CDF of the extreme Y_p is given by

$$\begin{aligned} F_{Y_p}(x) &= \Pr\left[\forall i : X_i \leq x\right] \\ &= \left(F_X(x)\right)^p. \end{aligned} \tag{2.8}$$

Similarly, the CDF of the extreme Y_1 is given by

$$\begin{aligned} F_{Y_1}(x) &= 1 - \Pr\left[\forall i : X_i > x\right] \\ &= 1 - \left(1 - F_X(x)\right)^p. \end{aligned} \tag{2.9}$$

The general case is given by the binomial distribution:

$$F_{Y_i}(x) = \sum_{j=i}^{p} \binom{p}{j} \left(F_X(x)\right)^j \left(1 - F_X(x)\right)^{p-j}, \tag{2.10}$$

see e.g. [129-Tho36]. This equation allows the construction of distribution free confidence intervals for quantiles. This will be discussed in the following section.

Equation (2.10) is used in [127-Str04] to calculate $F_{\hat{\Theta}}(\Theta)$, where $\hat{\Theta}$ is a quantile estimator and Θ is the expected value. $F_{\hat{\Theta}}(\Theta)$ is needed to construct a special case of min-max confidence intervals for quantiles. This approach is based

on multiple independent replications. Despite of this, Equation (2.4) and Equation (2.5) are still assumed for each replication, and if necessary, the data of each replication has to be transformed into an independent and identically distributed set of data. The advantage of independence of data of different replications is used only for the construction of the min-max confidence interval. The traditional way of confidence interval construction for quantiles will be discussed in the next section.

Using sorted random samples, Equation (2.6) can be changed to

$$\hat{F}_X(x) = \frac{1}{p} \min(i | x \geq Y_i) \tag{2.11}$$

with $1 \leq i \leq p$ and $\hat{F}_X(x) = 0$ for $x < Y_1$. The calculation of k points of $\hat{F}_X(x)$ from Equation (2.11) can be done in $\mathrm{O}(p \log p)$, because the data has to be sorted only once. Sorting can be done in $\mathrm{O}(p \log p)$. Then, all k points can be calculated in sorted order by a single linear search in the sorted sample of size p. This needs a run time of $\mathrm{O}(\max(k, p))$ and usually $k < p$ holds. The overall run time is, therefore, $\mathrm{O}(p) + \mathrm{O}(p \log p) = \mathrm{O}(p \log p)$. The range of $\hat{F}_X(x)$ is given by the distance $Y_p - Y_1$ of the two extremes. The error $\hat{F}_X(x) - F_X(x)$ is decreasing with increasing p, this will become clear in the discussions about quantiles of $F_X(x)$ in the next section.

2.2.2 Population and Sample Quantiles

Let x_q define a value in the range of X, so that $F_X(x_q) = q$. Therefore,

$$x_q = F_X^{-1}(q) = \inf\{x | F_X(x) \geq q\} \tag{2.12}$$

is the population quantile of order q, if $F_X(x)$ is continuous. If $F_X(x)$ is non-continuous this definition is ambiguous. The random interval $[Y_l, Y_u]$ is a distribu-

tion-free confidence interval for a population quantile, where

$$\begin{aligned} \Pr\left[Y_l \leq x_q \leq Y_u\right] &= \Pr\left[Y_l \leq x_q\right] - \Pr\left[Y_u < x_q\right] \\ &\geq \sum_{j=l}^{u-1} \binom{p}{j} q^j (1-q)^{p-j}. \end{aligned} \tag{2.13}$$

This equation can be derived of Equation (2.10) with $F_X(x_q) = q$, therefore, it is independent of the general form of $F_X(x)$. This property will be used extensively in later chapters, because it enables establishing a confidence interval for an unknown distribution.

The sample quantile $\hat{x}_q$ aims at estimating the population quantile x_q, when a certain value of q is specified. Common estimators are

$$\hat{x}_q = y_{\lfloor pq+1 \rfloor} \quad \text{or} \quad \hat{x}_q = y_{\lceil pq \rceil}, \tag{2.14}$$

where $\lfloor pq+1 \rfloor$ is the integer part of $pq+1$ and $\lceil pq \rceil$ is the smallest integer equal or greater than pq. However, many other estimators are known. For example the weighted sum of two neighbouring order statistics is another common estimator. In literature regarding simulation output analysis this is discussed e.g. in [140-WS95].

In later chapters we will extend beyond calculating x_q for a specified value of q. We rather try to estimate the whole CDF of a given measure on basis of several quantiles, assuming to be free to decide which values of q are appropriate and choosing them on the basis of the size p of a given sample. A sorted random sample naturally provides order statistics. Therefore, we are looking for the population quantile x_q that is represented by $\mathrm{E}\left[Y_i\right]$. $q = F_X(x_q)$ has to be estimated and x_q is given by y_i. We can see the dependence of q on the form of $F_X(x)$, thus, a general estimator of q for an unknown distribution is expected to be asymptotically (large sample size) unbiased only. In [33-DJ54] properties of the approximation

$$\text{(unknown case)} \quad \mathrm{E}\left[Y_i\right] \approx F_X^{-1}\left(\frac{i}{p+1}\right) \tag{2.15}$$

are discussed. The error decreases with growing sample size p and depends on derivatives of $F_X(x)$ as well as on the location of the quantile. Equation (2.15) suggests to estimate q by $\hat{q}_i = \frac{i}{p+1}$. This estimate is asymptotically unbiased for any form of $F_X(x)$ and it is the best for the uniform case. If the form of $F_X(x)$ is given specialised estimators are known. In [34-Dav70] is shown that for the exponential case

$$\text{(exponential case)} \quad \mathrm{E}\left[Y_i\right] \approx F_X^{-1}\left(\frac{i}{p+\frac{1}{2}}\right) \tag{2.16}$$

has better small sample properties than Equation (2.15). In this case $\hat{q}_i = \frac{i}{p+\frac{1}{2}}$ is a good estimate of q. For the normal case the small sample properties of

$$\text{(normal case)} \quad \mathrm{E}\left[Y_i\right] \approx F_X^{-1}\left(\frac{i-\frac{1}{2}}{p}\right) \tag{2.17}$$

are better than of Equation (2.15). Here, $\hat{q}_i = \frac{i-\frac{1}{2}}{p}$ is a good estimate of q. However, a general solution for the unknown case is given by Equation (2.15) and we can follow that $F_X(Y_i) \approx \hat{q}_i$, if p is sufficiently large. The unknown case is our main focus. In later chapters Equation (2.15) will be used to provide quantile estimators for the general case, i.e. for an unknown form of $F_X(x)$. If p is not sufficiently large the use of Equation (2.16) or Equation (2.17) will be necessary to obtain valid estimates. The large sample behaviour of estimators given in Equation (2.15), Equation (2.16) and Equation (2.17) is identical, so, they are asymptotically identical.

The difference between estimators like Equation (2.14) and estimators like Equation (2.15) is demonstrated by the following two cases:

(A) observation $\rightarrow$ rank $\rightarrow$ probability: Equation (2.15)

(B) observation $\leftarrow$ rank $\leftarrow$ probability: Equation (2.14)

In (A) the order statistic y_i (resp. observation) determines the rank i, then, the probability $F_X(y_i)$ is given by a simple sample proportion $\frac{i}{p+1}$, see Equation (2.15).

In this case $F_X(x)$ is computed for a given x. In (B) a probability q is given, e.g. specified by the analyst. The rank i is determined by $\lfloor pq + 1 \rfloor$ or $\lceil pq \rceil$, see Equation (2.14). The final estimate is the order statistic y_i at the determined rank i. In this case $F_X^{-1}(q)$ is computed for a given q. Case (B) applies the operators $\lfloor \cdot \rfloor$ or $\lceil \cdot \rceil$, which introduce additional bias due to discontinuous sample equations. In this research work observations are collected of simulation runs and their probabilities need to be calculated. Thus, case (A) is of higher interest and the estimators of Equation (2.15), Equation (2.16) and Equation (2.17) are applied, exclusively. We will compute $F_X(x)$ at a finite set of points $x_1, \cdots, x_p$. For this research work estimators like Equation (2.14) are not of further interest. This is contrary and novel compared to other methods of quantile analysis for simulation output data, see Section 2.4, because in other methods the analyst has to specify a certain q-quantile, or even a set of q-quantiles, for estimation.

2.2.3 Dependence of Sample Quantiles

Due to the relation $Y_i \leq Y_{i+1}$ order statistics form a dependent sequence. Let x_{q_1} and x_{q_2} be two quantiles of $F_X(x)$. The joint distribution of Y_i and Y_j, with $i < j$ and $x_{q_1} < x_{q_2}$, is

$$\begin{aligned} F_{Y_i Y_j}(x_{q_1}, x_{q_2}) &= \Pr\left[Y_i \leq x_{q_1}, Y_j \leq x_{q_2}\right] \\ &= \sum_{r=i}^{p} \sum_{s=\max(0,j-r)}^{p-r} \frac{p!}{r!s!(p-r-s)!} q_1^r (q_2 - q_1)^s (1 - q_2)^{p-r-s}, \end{aligned} \tag{2.18}$$

see [34-Dav70]. Therefore, two sample quantiles, which are estimated from the same sample, are dependent on each other.

The dependence of two quantiles taken from the same sample will be important in Chapter 4 and Chapter 6. To avoid quantile estimates that are located too close to each other and are, therefore, highly correlated we will introduce a concept of quantiles with disjoint confidence intervals in Section 2.3.

2.3 Disjoint Confidence Intervals

Two quantile estimates are correlated if the estimation is based on the same random sample. As the discussion of the previous section shows, the exact correlation depends on the underlying probability distribution. In [80-Ken40] the Gaussian approximation of distributions of quantiles is discussed. Using the Gaussian approximation for large samples $\mathrm{Var}\left[\hat{x}_q\right] = q(1-q)/p$ and $\mathrm{Cov}\left[\hat{x}_{q_1}, \hat{x}_{q_2}\right] = q_1(1-q_2)/n$ can be derived. These results can be used to calculate the asymptotic correlation

$$\begin{aligned} \frac{\mathrm{Cov}\left[\hat{x}_{q_1}, \hat{x}_{q_2}\right]}{\sqrt{\mathrm{Var}\left[\hat{x}_{q_1}\right]\mathrm{Var}\left[\hat{x}_{q_2}\right]}} &= \frac{q_1(1-q_2)}{\sqrt{q_1(1-q_1)q_2(1-q_2)}} \\ &= \sqrt{\frac{q_1(1-q_2)}{(1-q_1)q_2}}, \end{aligned} \tag{2.19}$$

which is valid for large samples and $0 < q_1 < q_2 < 1$. One can see, that for a fixed value of q_1 the asymptotic correlation tends to 0 for q_2 close to 1. On the other hand, for q_2 close to q_1 the asymptotic correlation tends to 1. This shows, that a mechanism is needed to control the correlation of two neighbouring quantile estimates. They should not be located too close to each other, because of high correlation. However, they should not be located too far away from each other, because then the estimation is too wasteful. One possible mechanism, that is not based on the Gaussian approximation and is also valid for small samples, is discussed next.

The discussion about disjoint confidence intervals of this section is partly published in [44-EMP07b]. Equation (2.13) allows to construct a confidence interval for a population quantile x_q based on two order statistics Y_l and Y_u. The probability $\Pr\left[Y_l \leq x_q \leq Y_u\right]$ can be calculated for arbitrary ranks $1 \leq l \leq u \leq p$, where p is the sample size and l and u define ranks representing lower and upper bounds. It is not necessary but could be desirable that x_q splits the confidence interval $[Y_l, Y_u]$ into two parts, so that half of the probability mass is in both parts.

Definition Let Y_c be an asymptotically unbiased estimate of x_q, i.e. $\mathrm{E}[Y_c] \approx F_X^{-1}(q) = x_q$. The confidence interval $\Pr[Y_l \leq x_q \leq Y_u] \geq 1-\alpha$ is *balanced* if

$$\begin{aligned} \Pr[Y_l \leq x_q \leq Y_c] &\geq \frac{1-\alpha}{2} \quad \text{and} \\ \Pr[Y_c \leq x_q \leq Y_u] &\geq \frac{1-\alpha}{2} \end{aligned} \tag{2.20}$$

hold.

This definition follows the concept of mid-p confidence intervals, see e.g. [16-BA95]. Other common approaches are to construct a confidence interval that has minimum width or that follows $x_q - Y_l = Y_u - x_q$. However, we construct the confidence interval on basis of Equation (2.20) because in the balanced case u and l can be calculated separately of each other.

Confidence intervals are always calculated for the deterministic value x_q. The "true" x_q is unknown, we estimate it by Y_c. A balanced confidence interval around Y_c can be calculated by determining q by Equation (2.15) in the general case, or by Equation (2.16) or Equation (2.17) in the exponential and normal case. Once q is determined we can initialise $l = u = c$ and calculate $\Pr[Y_l \leq x_q \leq Y_c]$ and $\Pr[Y_c \leq x_q \leq Y_u]$ by Equation (2.13) separately. l is decreased and u is increased until both conditions of Equation (2.20) are valid.

Let $\Pr[Y_{l_1} \leq x_{q_1} \leq Y_{u_1}] \geq 1-\alpha$ and $\Pr[Y_{l_2} \leq x_{q_2} \leq Y_{u_2}] \geq 1-\alpha$ be two balanced confidence intervals. As discussed, estimates of x_{q_1} and x_{q_2} are dependent, see Equation (2.18). However, by choosing disjoint confidence intervals, i.e. $u_1 \leq l_2$, we can ensure that at most $\frac{\alpha}{2}$ of the probability mass of both distributions overlap. If α is sufficiently small, e.g. $\alpha \leq 0.1$, high correlation between estimates of x_{q_1} and x_{q_2} can be avoided, compare with Equation (2.18).

To split the ordered sequence $\{Y_i\}_{i=1}^{p}$ into a maximum number of disjoint balanced confidence intervals an algorithmic approach is needed. A flowchart of this algorithm is depicted in Figure 2.1 for x_q, where $q > 0.5$. A flowchart for the

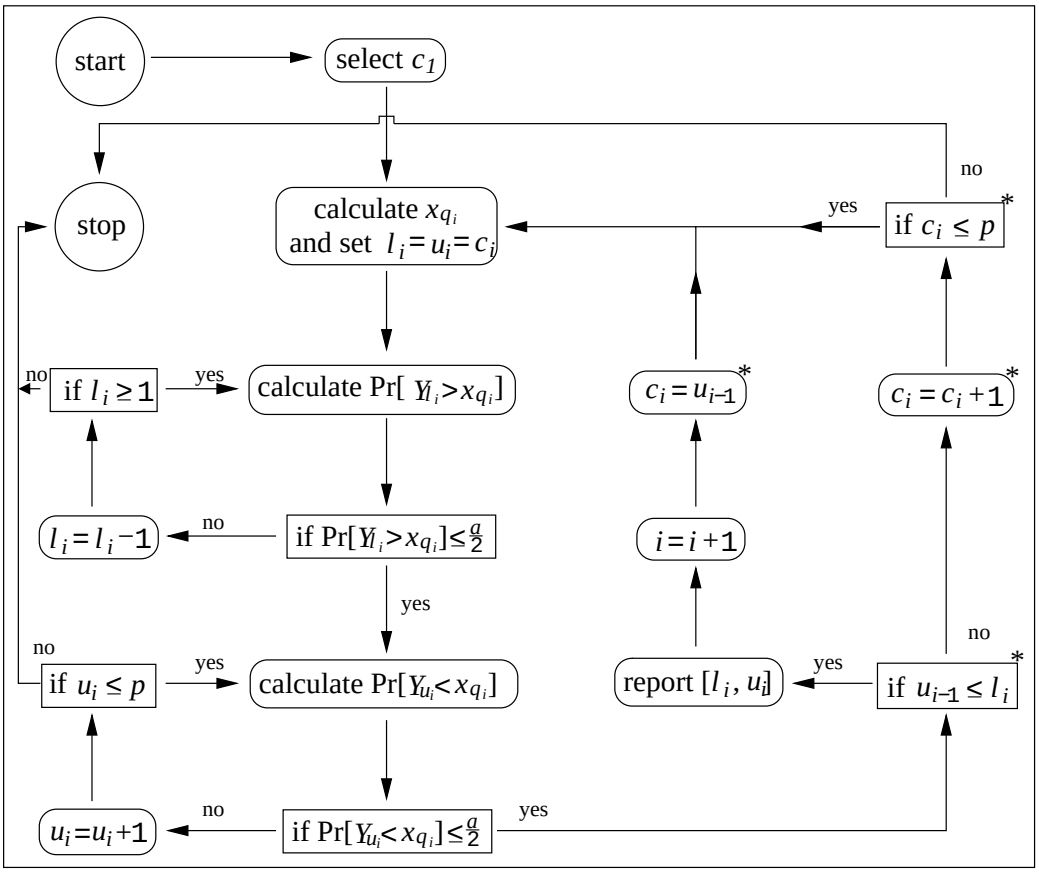


Figure 2.1: Flowchart of splitting $\{Y_i\}_{i=1}^{p}$ into a maximum number of disjoint balanced confidence intervals at confidence level $1 - \alpha$.

case $q < 0.5$ can be derived by substituting all components marked by a "*" with (if $l_{i-1} \geq u_i$), ($c_i = l_{i-1}$), ($c_i = c_i - 1$) and (if $c_i \geq 1$), respectively.

Let us assume that the size p of the random sample is odd and we start by selecting Y_{c_1} with $c_1 = \frac{p+1}{2}$. The assumption of an odd p assures that Y_{c_1} is an unbiased estimate of the median $x_{0.5}$ and all our further results are symmetric with centre $x_{0.5}$. The balanced confidence interval $[Y_{l_1}, Y_{u_1}]$ around Y_{c_1} at confidence level $1 - \alpha$ can now be calculated as described above. We start with $l_1 = u_1 = c_1$ and decrease l_1 and increase u_1 until Equation (2.20) holds. If this is successful the first confidence interval $[Y_{l_1}, Y_{u_1}]$ is given. To find a second confidence interval

$[Y_{l_2}, Y_{u_2}]$ with $u_1 \leq l_2$, we have to find the order statistic Y_{c_2} that estimates x_{q_2} so that $\Pr[Y_{l_2} \leq x_{q_2} \leq Y_{c_2}] \geq \frac{1-\alpha}{2}$. We start this search with $c_2 = u_1$. Now, q_2 can be determined by Equation (2.15) in the general case, or by Equation (2.16) or Equation (2.17) for the exponential or normal case. These equations describe how to find the unknown position of the quantile that is estimated by the given order statistic. After q_2 is estimated, the belonging Y_{l_2} can be calculated. If $u_1 > l_2$, this choice of c_2 should be rejected and $c_2 = u_1 + 1$, and so on, should be tested. The search can be stopped if $c_2 \leq p$ is violated, no more disjoint confidence intervals fit in the unprocessed area. If $u_1 \leq l_2$ holds, a valid choice of c_2 is found and additionally u_2 must be tested. If no $u_2 \leq p$ can be found the search can be stopped, otherwise another disjoint confidence interval is found.

Here, we have described the search for disjoint and balanced confidence intervals for x_q with $q > 0.5$. The search for x_q with $q < 0.5$ can be done analogously.

quantile (q)	rank (c)	lower bound (l)	upper bound (u)
0.007	7	2	14
0.023	23	14	34
0.047	47	34	61
0.077	77	61	95
0.114	114	95	135
0.157	157	135	181
0.206	206	181	232
0.259	259	232	287
0.316	316	287	346
0.376	376	346	407
0.438	438	407	469
0.5	500	469	531
0.562	562	531	593
0.624	624	593	654
0.684	684	654	713
0.741	741	713	768
0.794	794	768	819
0.843	843	819	865
0.886	886	865	905
0.923	923	905	939
0.953	953	939	966
0.977	977	966	986
0.993	993	986	998

Table 2.1: Disjoint and balanced confidence intervals for $p = 999$ and $\alpha = 0.05$.

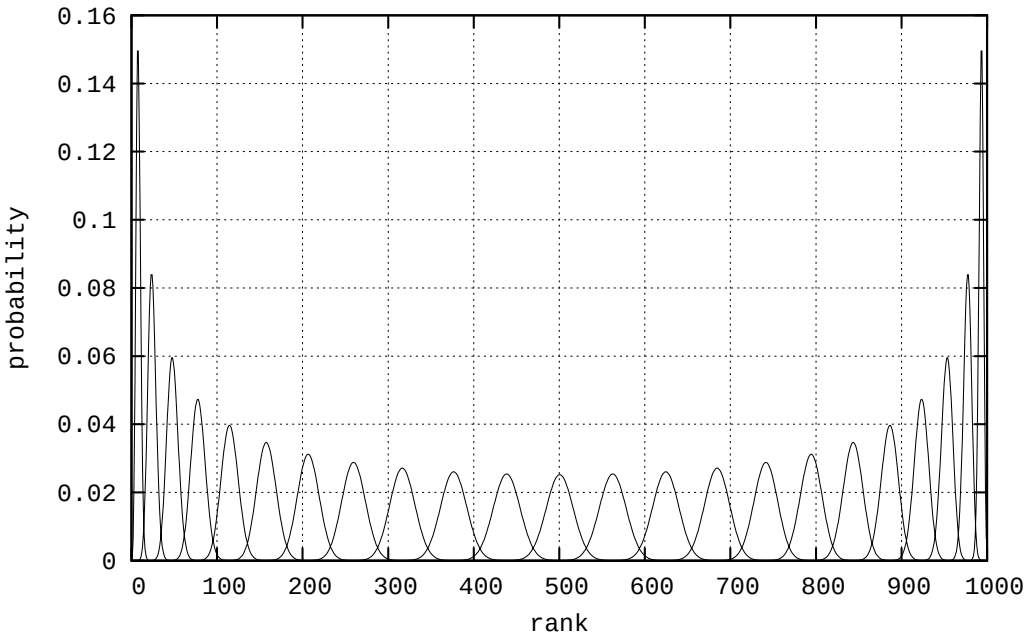


Figure 2.2: $\Pr[Y_i \leq x_q \leq Y_{i+1}]$ (ordinate), see Equation (2.13), versus rank i (abscissa) for q given by Table 2.1.

If the sample size p is even, we suggest not to start by estimating $x_{0.5}$, two different starting points for $q < 0.5$ and $q > 0.5$ could be used instead. For simplicity we described a linear search to find a valid value of c_2. Of course, this linear search could be replaced by a more efficient binary search within the unprocessed area. However, we do not think that efficiency of this approach is very important because the sample size p is usually not too large for fast execution. This way of splitting $\{Y_i\}_{i=1}^{p}$ into disjoint confidence intervals will be important later on, when several quantiles are used to interpolate a distribution function.

In Table 2.1 the results of our algorithm are shown in an example for $p = 999$ and $\alpha = 0.05$. The first column shows q, selected by Equation (2.15). The second column is the rank c of the belonging order statistic. The third and the fourth column are the bounds u and l of the balanced confidence interval. We can see that all confidence intervals are disjoint and that $\{Y_i\}_{i=1}^{p}$ is split into 23 parts. We

can even see that consecutive confidence intervals fit exactly to each other, so that $u_{i-1} = l_i$ holds for any i. The position and size of the confidence intervals are symmetric with centre at $q = 0.5$. Only the lowest and highest order statistics are not used in any confidence interval. The probability $\Pr[Y_i \leq x_q \leq Y_{i+1}]$, see Equation (2.13), is depicted against the rank i in Figure 2.2. For q we selected the values which are shown in Table 2.1. The density functions for low and high quantiles are asymmetric and have a small confidence interval. The confidence interval of the median is the largest and the density function is symmetric.

For $q = 0.5$ the distance $u - l$ is the largest, see Table 2.1. Because $l = 469$ and $u = 531$ the confidence interval of the median contains $u - l = 62$ order statistics. Using Equation (2.15) we receive a confidence interval size of $\frac{u-l}{p+1} = \frac{531-469}{999+1} = 0.062$ in the probability domain. The maximum error can be controlled by a specified threshold in the probability domain. It is possible to calculated how large the sample size p has to be to satisfy this threshold without the knowledge of Y_l and Y_u. A larger p leads to a smaller distance $u - l$. In this case only one random sample has to be collected. The confidence interval size in the domain of the measure can be calculated by $Y_u - Y_l$, which depends on the underlying distribution $F_X(x)$. In this domain, the confidence interval of the median is not necessarily the largest. Again, the maximum error can be controlled by a specified threshold in the domain of the measure. In this domain controlling the error is more difficult because the values of the order statistic Y_l and Y_u are needed. If $Y_u - Y_l$ is larger than the threshold a sample of larger size has to be collected. This should be done sequentially until $Y_u - Y_l$ is small enough to meet the threshold.

2.4 Quantiles of Simulation Output Streams

In previous sections we assumed random samples that are independent and identically distributed. However, in general the output stream of a simulation is a

stochastic process whose states at different time instances are dependent and not identically distributed. Thus, in general, Equation (2.4) and Equation (2.5) do not apply. In consequence, most other equations of previous sections do not directly apply. In the analysis of simulation output data special statistical methods are needed to estimate quantiles. This will be discussed in this section. However, Equation (2.4) and Equation (2.5) do apply across the collected data of multiple independent replications.

The most important property of a quantile estimate is its statistical accuracy with respect to an efficient calculation. The variance of a quantile estimator decreases as the number of observations increases. Random errors are caused by the stochastic variations of the simulation. They are caused by the fact that every simulation is like a statistical experiment. The next source of error is the bias of the estimator itself, often called the systematic error. This kind of error usually appears if assumptions about the analysed data hold only approximately or asymptotically. If both the variance and the bias tend to zero for large number of observations, the estimator is called consistent. More details about these statistical properties of quantile estimators can be found in [76-JC85]. There are further properties besides these statistical ones which characterise a suitable estimator. Storage requirements and calculation time are potentially important because usually a huge amount of output data needs to be processed to obtain trustworthy results. Therefore, not only the mathematical definition of the estimator, but also the way it is computed is of interest. Efficient data structures and algorithms are important. To guarantee a proper use of the estimator, even by inexperienced users, it is important that the quantile estimator is easy to understand and that the number of user-specified parameters is small, preferably zero. A classification of these properties is given e.g. in [63-GS97] for the general problem of estimating standard error.

2.4.1 Single Quantile

The estimation of a single quantile of the steady state distribution, when simulating a single instance of a time-stationary process, is considered by Iglehart, Seila, Heidelberger and Lewis, Jain and Chlamtac, Chen and Kelton (see e.g. [74-Ige76], [119-Sei82], [71-HL84], [76-JC85], [25-CK99]). The methods of Iglehart and Seila are limited to regenerative processes. The subdivision of the output data into its regenerative cycles is a natural way to overcome the problem of autocorrelation. The method of Seila extends the method of Iglehart by grouping the regenerative cycles into batches. The number of parameters which have to be specified by the user is reduced by this batching approach to one parameter: the batch size. However, the determination of the batch size is a difficulty common to every batching approach; it is difficult for an inexperienced user to choose an appropriate value. The method of Heidelberger and Lewis addresses the problem of quantile estimation in dependent sequences. Their method is not limited to regenerative processes, which is an important improvement for the analysis of simulation output data. The point estimate based on ordered data is still valid in the dependent case, but its variance may be inflated leading to a larger interval estimate. Two basic solutions are given. On the one hand, the higher variance can be calculated directly with the spectral method (see [72-HW81]). On the other hand, the data can be transformed to almost independent data by using a batch means method (see e.g. [52-FY97]). Because this method is closely related to mean value analysis, we will discuss it in more detail at the end of this section. The method of Jain and Chlamtac uses a completely different kind of quantile estimator. Their estimator is based on markers, which are adjusted when collecting new observations. This is done by a piecewise-parabolic interpolation. Because of this interpolation, this method is not recommended for quantile estimation of discontinuous distribution functions. The estimator seems to be quite complicated compared to the usual estimators based on ordered data. However, the principal

advantage is that the method requires only a constant (and small) amount of memory. Chen and Kelton describe a method that estimates a quantile by focusing on observations which are located in the neighbourhood of this quantile. Their method is sequential to ensure an accurate final estimate. However, the quality of this method has not been exhaustively studied yet.

A method for quantile estimation in finite-horizon simulation is described in [7-AW95] and [8-AW98]. This method is based on multiple replications of the finite-horizon simulation. These replications are dependent on each other because negative correlation is introduced into their streams of input random numbers to reduced variance. Avramidis and Wilson propose that this approach yields improvements under special assumptions (see also [77-JFX03]). However, more extensive experimental evaluation of the proposed quantile estimators is needed.

The estimation of one single quantile is usually done to analyse the tail behaviour of a distribution. In this case typically the 0.95-quantile (resp. 0.05-quantile) is estimated. For more extreme quantiles than this it might be more appropriate to use rare event simulation. However, sometimes the median (0.5-quantile) is estimated instead of the mean value, because the median is more robust against outliers.

Here, we will discuss the *Method of Heidelberger and Lewis*, which is proposed in [71-HL84], in more detail. It is closely related to mean value analysis and is, therefore, a candidate for further extensions. In Chapter 6 we will refer back to this section. This method of quantile estimation of simulation output data is maybe the first approach that is suitable for dependent, but identically distributed, sequences $X_1, X_2, \ldots$, as this is the situation in steady state phase. The maximum transformation $M = \max(X_1, X_2, \ldots, X_v)$ is used to estimate a quantile $x_q = F_X^{-1}(q)$, with $q > \frac{1}{2}$. For $q < \frac{1}{2}$ a minimum transformation can be defined analogously. The CDF $F_M(x)$ can be derived by Equation (2.8):

$$F_M(x) = \Pr[M \leq x] = (F_X(x))^v. \tag{2.21}$$

The probability of quantile x_q is, therefore, $F_M(x_q) = q^\upsilon$. In [71-HL84] is pointed out that choosing υ so that $q^\upsilon \approx 0.5$ is desirable. In this way the problem of estimating an extreme quantile is reduced to the estimation of the median. The drawback of this transformation is that the variance of the estimator is increased.

The original output process X_1, X_2, ..., $X_{m\upsilon}$ of fixed size $m\upsilon$ is partitioned and transformed into $M_i = \max_{1\leq k\leq \upsilon}(X_{i+(k-1)m})$, so that we receive M_1, M_2, ..., M_m. Heidelberger and Lewis point out, that the purposes of the transformation are

- a reduction of the sample size,
- estimating the median rather than an extreme quantile,
- to possibly reduce correlation.

The reduction of correlation depends on the particular output process. The sequence of random numbers cannot be randomised, because this will hide the original autocorrelation structure.

Let $M'_1 \leq M'_2 \leq \ldots \leq M'_m$ be the sorted sequence of M_1, M_2, ..., M_m. A point estimate of x_q is given by

$$\hat{x}_q = M'_{\lfloor mq+1 \rfloor}, \tag{2.22}$$

compare with Equation (2.14). The estimation of an interval estimate is still not straight forward. Because the sequence M_1, M_2, ..., M_m is correlated, the estimation of the variance involves advanced techniques. A spectral method is suggested, as it is done in [72-HW81] for mean value analysis. As additional alternatives variations of the batch means method, see e.g. [52-FY97], are used.

Heidelberger and Lewis recommend that at least 10% precision be obtained, although this choice does not guarantee valid confidence interval coverage. The batch mean methods seem to save more storage than the spectral method. Further advantages are that they are conceptually simpler and more readily adaptable

to sequential methods. However, they require a more stringent independence assumption than the spectral method, according to Heidelberger and Lewis. The problem of the presence of an initial transient, i.e. the sequence X_1, X_2, ..., X_{mv} is not identically distributed, is not addressed in [71-HL84]. Furthermore, the determination of a valid batch size for the batch mean methods is not addressed, either.

2.4.2 Several Quantiles

If the analyst is interested in the complete distribution function of a performance measure the estimation of several quantiles is useful, because the quantiles describe the probability distribution at special points. The estimation of several quantiles of the steady state distribution is addressed by Raatikainen, see for example [108-Raa87]. The method of Jain and Chlamtac is extended by introducing additional markers to estimate more quantiles. The adjustment of the markers is done in the same way as before. An investigation of the variance of this method is given in [109-Raa90]. A different approach is proposed in [110-Raa95]. In previous publications the location in the range of the measure is estimated for a fixed probability. Here, the probability of a predefined "category" of the range of the measure is calculated. The most obvious "category" is maybe $X \leq x$ resulting in a point and interval estimate of $F_X(x)$. This method is well tested and known to give statistically accurate results, therefore, we will discuss it in detail at the end of this section.

One of the main difficulties in quantile estimation is the high computational effort and the large amount of storage needed to order the observations. Therefore, Heidelberger and Welch reduce the sample size by a maximum transformation (see [71-HL84]). Jain, Chlamtac and Raatikainen go further and avoid sorting the output data by using an interpolation. In recent publications of Hashem, Schmeiser and Wood (see [68-HS94] and [139-WS94]) or Chen and Kelton (see

[26-CK01] and [24-Che02]) quantile estimators based on order statistics have become popular again. This may be due to increased memory and processor speeds making these methods more practical. Wood and Schmeiser describe a batching method for quantiles which is similar to batch means and consider different quantile estimators, all based on ordered observations. The batch statistic is given by one of four quantile estimators, which are all based on ordered observations. Again, the difficulty is how to chose an appropriate batch size.

In [26-CK01] the previous method of estimating a single quantile is extended to the problem of estimating several quantiles. Again, the extended method is sequential as the previous version. In [27-CK06] the coverage of both, the single and the multiple quantile estimator are assessed. The coverage of the single quantile estimator is even higher than expected. This estimator also reduces the amount of data that needs to be stored. However, the average run length of all experiments is below $12.5 \cdot 10^6$ observations. Storing this data takes about 200 MB of memory, assuming 16 bit numbers, and is no problem for modern computers. Even sorting of this data should not take long if efficient data structures are used and the data is sorted by merging small samples into the already sorted large sample. Chen and Kelton state that savings in storage and sorting are substantial. This statement is not supported by their experimental results. For some quantiles of correlated data the coverage of the multiple quantile estimator is not as expected. This might indicate that the runs-up test (see [81-Knu98]), which is used to transform correlated data into quasi-independent data, is not the best choice. Furthermore, the issue of correlation between estimated quantiles of the same sample is not addressed.

Two different density estimators are described in [28-CK06]. One is based on histogram estimation, which is closely related to quantile estimation. The other one is based on the use of a kernel function. By experiment Chen and Kelton show that the histogram density estimator is superior because the coverage of estimated confidence intervals is better. They conclude that the histogram procedure is more

suitable as a generic density estimation procedure since it requires less computation and delivers a valid confidence interval.

The most recent *Method of Raatikainen*, see [110-Raa95], aims at the estimation of the probability $q_k = \Pr[X \in \mathcal{C}_k]$ of a predefined "category" $\mathcal{C}_k$ of a random variable X in steady state. Here, we will discuss this method in more detail because it is probably the most studied estimator. It will be used for comparison in later chapters of this thesis. By defining $\mathcal{C}_k = [-\infty, x_k]$ this method estimates $q_k = F_X(x_k)$ for $1 \leq k \leq m$. Raatikainen remarks that this method is suitable for $5 \leq m \leq 25$ estimates. The main idea is to use the arc sine transformation to determine interval estimates for all q_k.

Let $X_1, X_2, \ldots, X_n$ be the simulation output at a temporary simulation horizon n. An estimate of q_k is given by

$$\hat{q}_k = \frac{1}{n} \sum_{j=1}^{n} I_{k,j}, \tag{2.23}$$

where

$$I_{k,j} = \begin{cases} 1 & \text{if } X_j \in \mathcal{C}_k, \\ 0 & \text{else.} \end{cases} \tag{2.24}$$

is a binary function. Note the similarities of Equation (2.23) and Equation (2.6). Raatikainen points out that a confidence interval for $\hat{q}_k$ can be calculated by

$$\begin{aligned} \hat{q}_{l_k} &= \left(\sin \left(\max \left(0, \sin^{-1} \left(\sqrt{\hat{q}_k} \right) - \frac{\delta_k}{2} \right) \right) \right)^2 \quad \text{and} \\ \hat{q}_{u_k} &= \left(\sin \left(\min \left(\frac{\pi}{2}, \sin^{-1} \left(\sqrt{\hat{q}_k} \right) + \frac{\delta_k}{2} \right) \right) \right)^2 \end{aligned} \tag{2.25}$$

for the lower and upper bound, where δ_k is the desired halfwidth specified by the analyst. We receive the confidence interval $[\hat{q}_{l_k}, \hat{q}_{u_k}]$. According to Raatikainen, the following relation of the confidence level $\hat{\alpha}_k$ holds:

$$\delta_k = F_t^{-1} \left(1 - \frac{\hat{\alpha}_k}{2}; \nu_k \right) \cdot \frac{s_{\hat{q}_k}}{\sqrt{\hat{q}_k (1 - \hat{q}_k)}}, \tag{2.26}$$

where $F_t(x;\nu_k)$ is the cumulative probability of the t-distribution with ν_k degrees of freedom and $s^2_{\hat{q}_k}$ is an estimate of $\mathrm{Var}\,[\hat{q}_k]$. So, the confidence level $\hat{\alpha}_k$ of the confidence interval $[\hat{q}_{l_k}, \hat{q}_{u_k}]$ is given by

$$\hat{\alpha}_k = 2F_t\left(-\delta_k \frac{\sqrt{\hat{q}_k(1-\hat{q}_k)}}{s_{\hat{q}_k}}; \nu_k\right). \tag{2.27}$$

In Equation (2.26) and Equation (2.27) the estimate $s^2_{\hat{q}_k}$ is needed. As Raatikainen points out, there are various different ways of estimating $\mathrm{Var}\,[\hat{q}_k]$ like batch mean, see e.g. [50-Fis78], or spectral analysis, see e.g. [72-HW81].

Raatikainen defined a stopping criterion for sequential simulation that is based on Bonferroni's inequality:

$$\sum_{k=1}^{m} \hat{\alpha}_k \leq \alpha, \tag{2.28}$$

where α is the combined confidence level with

$$\Pr\left[q_1 \in [\hat{q}_{l_1}, \hat{q}_{u_1}] \wedge \cdots \wedge q_m \in [\hat{q}_{l_m}, \hat{q}_{u_m}]\right] \geq 1-\alpha. \tag{2.29}$$

The simulation can be stopped if Equation (2.28) is fulfilled. For more details about this stopping criterion and an alternative stopping criterion see [110-Raa95].

Our focus is to estimate $F_X(x)$ on basis of several quantiles. This may involve the calculation of much more than 25 quantiles, which is the highest recommended number of estimates for this method. The reason for this restriction of Raatikainen's method is the use of Bonferroni's inequality in Equation (2.28). If m is large the corresponding $\hat{\alpha}_k$ have to be very small. Coverage of the corresponding confidence intervals might shrink faster than expected.

The estimates $\hat{q}_k$ are based on $I_{k,j}$, which are derived from the same sequence $X_1, X_2, \ldots, X_n$. Therefore, all $\hat{q}_k$ are correlated among each other. Especially for two neighbouring estimates higher correlation can be expected. Possibly a recommendation is needed, how to choose the k "categories" $\mathcal{C}_k$ optimally. Depending on the purpose, an automatic selection of a set of $\mathcal{C}_k$ would be desirable.

2.5 Summary

In this chapter we showed various characteristics of simulation models, output processes and estimators. We pointed out that estimating a set of quantiles is a promising way to approximate the underlying probability distributions. For this reason we discussed basic mathematics of quantile estimation of an independent and identically distributed random sample. We also introduced a way of selecting a set of quantiles with disjoint confidence intervals to avoid high correlation between the estimates. The special situation of quantile estimation of simulation output data is surveyed. Methods are explained in detail, which will be used for comparison and for further extensions. These methods are probably the most studied estimators with good statistical performance. In further chapters we would like to extend and improve on these existing methods by transferring them to the simulation scenario of independent replications. The discussion of the advantages of independent replications is done in the next chapter.

Chapter 3

Parallel Simulation Scenarios

In this chapter we discuss parallel simulation scenarios in general. Parallelisation of simulation can be done at various levels, for example we can split the simulation program into functional units, decompose the simulation model into almost independent submodels, parallelise the generation process of output data, or parallelise the analysis of output data, see [111-RW89]. Here we want to describe the kind of parallelisation we require, its advantages and differences to other parallel simulation scenarios.

An early approach for parallelising simulation was to distribute functional units to parallel engines. Functional units are those components of a simulation, which are not part of the model itself, for example the pseudo-random number generator. Because the number of functional units is usually small, the degree of parallelisation and, thus, the achievable speedup is very limited. Therefore we do not consider this kind of parallelising any further.

3.1 Decomposition of Simulation Models

One common way of parallelising of simulation is to distribute the simulation model itself, which is commonly referred to as parallel discrete event simulation, see [53-Fuj90]. This assumes that the model is decomposable into submodels. The number of possible submodels determines how many parallel engines can be

used to make a given simulation more efficient. This approach is attractive if the memory requirement of the simulation is too high for a single computer. However, as well as the need for a decomposable model, there is one main disadvantage. The communication between the submodels can be difficult. To assure that the simulation of submodels is synchronised, avoiding causality errors, various complex techniques have been introduced. These range from the conservative to the optimistic class of techniques. The conservative approach strictly avoids causality errors due to submodels operating at different model times. Optimistic approaches try to detect causality errors and rollback to a recovery state. In this scenario the degree of parallelising is limited by the model itself, because decomposing the model requires almost independent submodels.

3.2 Classical Independent Replications

In contrast to decomposition of the simulation model there are methods who start replications of the same simulation model using different random numbers. Note, that the use of replications does not exclude the possibility to distribute each simulation model. In Section 2.1.2 we stated that the output data of a simulation run forms a sequence of random variables $\{X_i\}_{i=1}^{\infty}$. In general, the CDF $F_{X_i}(x)$ depends on the initial state of the simulation model, especially for small i. Let $\{x_i\}_{i=1}^{\infty}$ be a realisation of $\{X_i\}_{i=1}^{\infty}$. This means that $\{x_i\}_{i=1}^{\infty}$ contains the observations of one given simulation run. These observations also depend on the initial state of the model. In stochastic computer simulation pseudo-random number generators are used to introduce randomness into the simulation model. p simulation runs of the same model with the same initial state but with different seeds

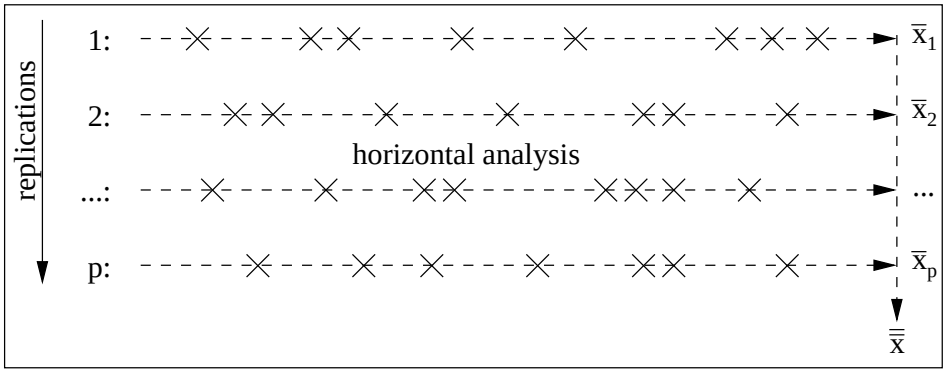


Figure 3.1: Mean value estimation by horizontal analysis using the IR scenario. The mean $\bar{x}_j$ of each replication can be calculated. The set of all $\bar{x}_j$ form an independent and identically distributed sample. On basis of this sample an overall mean $\bar{\bar{x}}$ and its confidence interval can be calculated.

result in p different independent sequences of observations:

$$\begin{aligned}
\text{Replication } 1: &\quad x_{1,1}, \quad x_{1,2}, \ldots \\
\text{Replication } 2: &\quad x_{2,1}, \quad x_{2,2}, \ldots \\
&\quad \ldots \\
\text{Replication } p: &\quad x_{p,1}, \quad x_{p,2}, \ldots
\end{aligned}$$

All p sequences are realisations of $\{X_i\}_{i=1}^{\infty}$ and are called replications. To denote the set of p sequences we will use the notation $\left\{\{x_{j,i}\}_{i=1}^{\infty}\right\}_{j=1}^{p}$. $x_{j,i}$ is the ith observation of the jth replication. n_j denotes the number of observations collected in each replication. Dependencies between the components of $\{x_{j,i}\}_{i=1}^{n_j}$ and $\{x_{j',i}\}_{i=1}^{n_{j'}}$, where $j \neq j'$, can be effectively excluded by using non overlapping streams of pseudo-random numbers in each replication.

The use of independent replications (IR) to estimate the steady state mean is discussed in [85-LK00]. Each sequence $\{x_{j,i}\}_{i=1}^{\infty}$ is used to calculate an estimate $\bar{x}_j$:

$$\bar{x}_j = \frac{1}{n - l_E} \sum_{i=l_E+1}^{n} x_{j,i}. \tag{3.1}$$

Here, l_E is the truncation point to distinguish between the transient and the steady

state phase in mean value analysis and $\forall(j) : n_j = n$ is assumed. With regard to the index i we will call this approach horizontal analysis of the output data, as depicted in Figure 3.1. Then, the overall mean is given by

$$\bar{\bar{x}} = \frac{1}{p} \sum_{j=1}^{p} \bar{x}_j. \tag{3.2}$$

The estimates $\bar{x}_1$, $\bar{x}_2$, ..., $\bar{x}_p$ are independent of each other and the variance of the overall mean can be calculated by standard methods. The variance and the assumption of a normal or a Student's t-distribution leads to an interval estimates of the steady state mean. For large p (≥ 100) normality can be assumed, for small p the Student's t-distribution is chosen.

The parallel simulation scenario of independent replications shortens the time needed for collecting p independent sequences, due to parallel generation of output data. In general, the degree of parallelisation is not limited. Replications can easily be run in parallel on many processing engines. One of the main advantages of running parallel replications is the increased speed in collecting observations, which is linearly bounded according to p. Practical issues of running replications in parallel, such as scheduling policies and the effects of changes in workload, are discussed for example in [89-Lin95]. In [69-Hei86] the statistical properties of estimates obtained by a distributed simulation model and parallel independent replications are compared. One conclusion is that "if the run length is long or if the initial transient is weak, then replications will be statistically more efficient than distributed simulation (distribution of the model) in estimating steady state quantities". However, in our understanding both the distribution of a model, and performing parallel replications, are two different techniques which do not exclude each other. Furthermore, in [70-Hei88] the issue of replications with too short run length is discussed. It is pointed out that estimates based on a very large number of too short replications almost certainly lead to biased estimates. This result is supported by [138-Whi91] and additionally it is stated, that "it usually is

more efficient to make one long run than to make independent replications", but "it usually does not matter much". The issue of steady state analysis using independent replications is considered in [61-GH92a]. It is pointed out that special care needs to be taken in order to obtain estimators with the proper convergence behaviour, and that the number of replications, the length of the replications and the length of the initial transient need to be controlled in order to produce valid confidence intervals for steady state parameters.

In this section we discussed classical independent replications, where analysis is usually based on samples of fixed size. We distinguish two additional versions of independent replications which operate without a fixed sample size and are applicable in sequential simulation. In the scenario of asynchronous independent replications each replication is of different length. In the scenario of synchronous independent replications all replications have the same length at all intermediate stages of analysis. These two scenarios are discussed next.

3.3 Asynchronous Independent Replications

In asynchronous replications each replication runs at its own individual speed, thus, each replication may produce a different number observations. An extension of the independent replications scenario is the use of multiple independent replications in parallel (MRIP), as introduced and discussed in [105-PYM94] and [100-Paw00]. In addition to parallel generation of output data the analysis is partly distributed. In the MRIP scenario replications are asynchronous. They run at their own speed and do not wait for slower replications. Therefore, we can expect that replications deliver output data at different rates, i.e. if $j \neq j'$ then $n_j \neq n_{j'}$. When using asynchronous replications, it is advisable to perform an asynchronous analysis of each replication to use the full potential of parallel computation. This means that all $\{x_{j,i}\}_{i=1}^{n_j}$ are analysed separately and deliver local estimates. All local estimates of the p replications can be combined to a global estimate, as de-

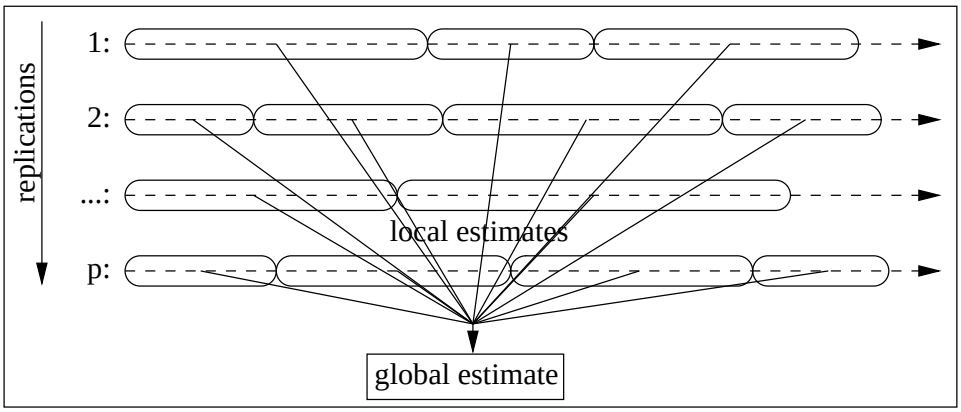


Figure 3.2: In the MRIP scenario the global estimate is calculated on basis of local estimates. Every time a replication reaches a checkpoint analysis of the output data of this replication is done. Local estimates with adequate statistical properties are transmitted to a global analyser that calculates a global estimate.

picted in Figure 3.2. With regard to the index i a horizontal analysis of the output data is done.

This approach is very time efficient because every engine can run at its own speed. In [102-PM01] the increase in speed is analysed in more detail. It is pointed out that the speedup of parallel replications follows Amdahl's law ([4-Amd67]). The speedup is not always linear because parts of the simulation experiment cannot be run in parallel, e.g. the generation of seeds for the pseudo-random number generator. Thus, the mean speedup $\bar{S}$ of sequential stochastic simulation using MRIP depends on p as:

$$\bar{S} = \begin{cases} \frac{1}{\bar{f}+(1-\bar{f})/p}, & \text{for} \quad p \leq \lceil \frac{(1-\bar{f})\bar{N}_{\min}}{\bar{D}} \rceil; \\ \frac{\bar{N}_{\min}}{\bar{f}\bar{N}_{\min}+\bar{D}}, & \text{for} \quad p \geq \lceil \frac{(1-\bar{f})\bar{N}_{\min}}{\bar{D}} \rceil, \end{cases} \tag{3.3}$$

where $\bar{N}_{\min}$ is the mean number of observations needed for stopping simulation when a single simulation engine is used, $\bar{f}$ is the mean fraction of the $\bar{N}_{\min}$ observations that cannot be produced in parallel (e.g. the initial transient phase), and $\bar{D}$ is the mean distance between checkpoints. This law, formulated in [102-PM01], has been called a truncated Amdahl law of MRIP. It is assumed that replications do not need any supervision or synchronisation and the p simulation engines make

exactly the same contribution to a given simulation. One can see that $\bar{S}$ is a linear function of p only if all observations can be cooperatively produced in parallel, i.e. $\bar{f} = 0$, and the maximum of $\bar{S}$ strongly degrades as $\bar{f}$ increases.

The speedup $\bar{S}$ is given by Equation (3.3) in the asynchronous case. However, since a homogeneous set of processors operating as simulation engines is assumed, $\bar{S}$ determines the upper bound of mean speedup. This approach is fault tolerant with regard to p, the number of replications. If one replication is lost, e.g. one of the engines is disconnected for any reason, the simulation experiment can be continued on basis of the remaining $p - 1$ replications. An implementation of the asynchronous MRIP scenario is AKAROA-2 ([47-EPM99]), which is a fully automated simulation tool designed for running distributed stochastic simulations in a local area network.

3.4 Synchronous Independent Replications

In this section we would like to describe the parallel simulation scenario that will be used in later chapters. This scenario is mainly focused on the estimation of distribution functions, and provides some speedup compared to a single simulation run. It is related to the MRIP and ID scenario. However, the analysis of the data is done in a vertical way across replications. This will be explained later.

We consider parallel replications of one simulation run. In this way the generation of output data can be increased compared to one long simulation run: instead of collecting just one value at each observation index, a whole sample of p observations can be collected. The statistical advantage of having an independent and identically distributed random sample of X_i due to the use of parallel replications seems to be a good starting point for quantile estimation.

In the MRIP scenario local estimates make a distributed analysis possible. Unfortunately it is not always possible to give a local estimate on basis of one replication for any kind of performance measure. In mean value analysis a local

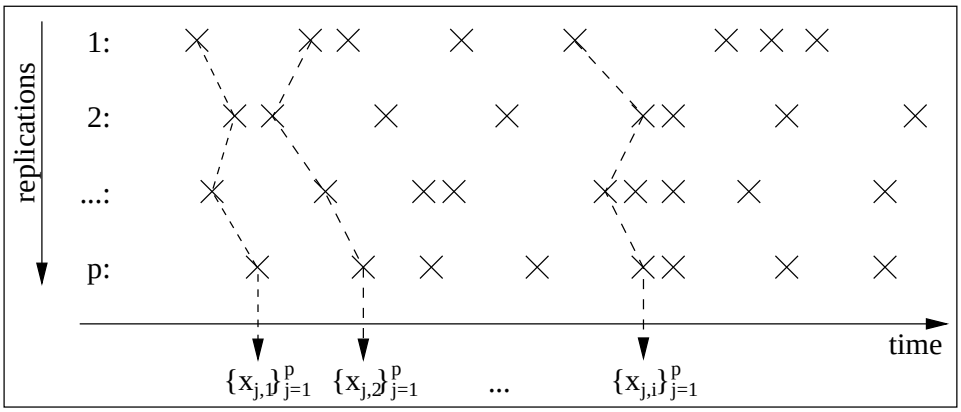


Figure 3.3: Synchronised data collection of independent replications and vertical analysis of output data. The ith observations of each replication are combined to an independent and identically distributed sample. On basis of this sample statistical properties of X_i can be derived.

estimate is basically just the average of a substream of $\{x_{j,i}\}_{i=1}^{n_j}$. When estimating quantiles the local estimate is difficult to find on basis of $\{x_{j,i}\}_{i=1}^{n_j}$, because standard quantile estimators are usually based on independent and identically distributed random samples (see Section 2.2) and techniques for the dependent case have to be applied (see Section 2.4). Here, a quantile can be estimated on basis of $\{x_{j,i}\}_{j=1}^{p}$, which is not a local estimate. $\{x_{j,i}\}_{j=1}^{p}$ (column) is an independent and identically distributed sample, whereas $\{x_{j,i}\}_{i=1}^{n}$ (row) is autocorrelated and may not be identically distributed. To be able to perform this kind of vertical analysis, synchronous replications are needed so that $\forall(1 \leq j \leq p) : n_j = n$ at any time. The synchronisation of replications can be done in a relative simple way by using a buffer for each replication. The set of output data can be sequentially extended by simply accessing the next observation index $n + 1$. The final length of a simulation experiment is not determined because the current simulation horizon n can always be increased. Using this kind of synchronisation the issue of too short replications, as discussed in [70-Hei88] and [138-Whi91], does not occur.

In the synchronous case all replications can run only as fast as the slowest engine, but still, a new sample $\{x_{j,n+1}\}_{j=1}^{p}$ is available instantaneously when each

of p simulation engines generates just one observation. The communication overhead is also an issue for quantile estimation because a single engine is not able to give a local quantile estimate. Details of the quantile estimators proposed in this thesis are given in Chapter 4. In Equation (3.3) the mean distance between checkpoints $\bar{D}$ of analysis is assumed to be constant. In Chapter 6 we will see that for the synchronous case and quantile estimation a geometrically growing distance is advisable to reduce too many unnecessary calculations. Synchronous replications have some mathematical advantages. At every observation index i we receive $\{x_{j,i}\}_{j=1}^{p}$. With regard to the index j this approach represents a vertical analysis of the output data, as depicted in Figure 3.3. This is a new approach because both the IR and MRIP scenario perform horizontal analysis. If the replications are independent of each other, the observations $\{x_{j,i}\}_{j=1}^{p}$ at a fixed observation index i represent an independent and identically distributed random sample of X_i. This is a very important feature of the analysis of synchronous replications because it enables new possibilities for output analysis. $\{x_{j,i}\}_{j=1}^{p}$ fulfils all assumed preconditions of Section 2.2. If p is sufficiently large, quantiles $F_{X_i}^{-1}(q)$ can be estimated by standard methods and the empirical CDF $\hat{F}_{X_i}(x)$ can be calculated for any i. We can get an impression of how $F_{X_i}(x)$ is developing with growing i. In later chapters this knowledge of $F_{X_i}(x)$ will be used intensively: In Chapter 4 methods for depicting the evolution of quantiles of $F_{X_i}(x)$ will be described, and in Chapter 5 homogeneity tests are used to find the steady state phase of simulation. The use of synchronous replications greatly assists the analysis of $F_{X_i}(x)$ developing over i.

3.5 Random Numbers

The generation of multiple streams of pseudo-random numbers with adequate statistical properties is very important for all scenarios using replications. Dependencies between the components of $\{x_{j,i}\}_{i=1}^{n_j}$ and $\{x_{j',i}\}_{i=1}^{n_{j'}}$, where $j \neq j'$, can

be avoided by using non overlapping streams of pseudo-random numbers in each replication. We use pseudo-random number generators where the cycle of pseudo-random numbers can be split into substreams. For a survey on this topic see e.g. [104-PSM06]. For experiments in later chapters we will use the pseudo-random number generators which is described in [87-LSCK02]. This generator belongs to the class of combined multiple recursive generators and is based on 2 components, $g_{1,k}$ and $g_{2,k}$, each of order 3. They evolve according to the linear recurrences

$$\begin{aligned} g_{1,k} &= (1403580 g_{1,k-2} - 810728 g_{1,k-3}) \mod m_1 \quad \text{and} \\ g_{2,k} &= (527612 g_{2,k-2} - 1370589 g_{2,k-3}) \mod m_2, \end{aligned} \tag{3.4}$$

where $m_1 = 2^{32} - 209$ and $m_2 = 2^{32} - 22853$. The output is defined by

$$u_k = \frac{(g_{1,k} - g_{2,k}) \mod m_1}{m_1 + 1}. \tag{3.5}$$

The special case $u_k = 0$ is avoided by returning $\frac{m_1}{m_1+1}$ instead. The period length is approximately 2^{191} numbers and substreams of 2^{76} numbers are available. The size of these substreams should be larger than any possible n_j. In this case the replications can be said to be independent of each other and are realisations of the same stochastic process $\{X_i\}_{i=1}^{\infty}$.

In [86-LS07] (published during review process of this thesis) a test environment for pseudo random number generators is introduced and the performance characteristics of more than 90 common random number generators is examined. It is stated that default generators of many popular software programs, e.g. Excel, MATLAB, Mathematica, fail several tests miserably. For simulation the random number generator with multiple streams and substreams is recommended, which is previously described in this section and introduced in [87-LSCK02]. For all simulation experiments of this research work we exclusively applied this generator.

3.6 Summary

In this chapter we discussed basic scenarios of parallel simulation and compared the main ways of parallelising a simulation experiment. If the replications are synchronised output analysis can be done for every synchronisation point, i.e. for every observation index i, separately. The random sample of one observation index i is independent and identically distributed and enables the use of advanced statistical methods. Methods of this kind will be demonstrated in the following chapters.

Chapter 4

Time Evolution of Quantiles

An extension of the problem of estimating several quantiles at a given time point or in steady state, is analysis of the time evolution of these quantiles as the simulation progresses. This provides deeper insight into the transient behaviour of the system of interest. In steady state simulation this can help to verify if a steady-state phase exists, i.e. that the probability distribution function of the analysed performance measure is converging to its steady-state form. Parts of the discussion and results of this chapter are published in [40-EMP05a], [42-EMP06]. An application in the analysis of time dependent file popularity in Peer-to-Peer networks is published in [17-BEPS07].

In applications, finite-horizon simulation is frequently used to examine a given process with a certain initial state. In this case the transient behaviour of the system is the central point of interest. Again, the estimation of several quantiles over time provides a deeper insight than mean value analysis only. The estimation of several quantiles in possibly time non-stationary processes has had limited attention.

The methods of simulation output data analysis, which are discussed in this section, are based on synchronous replications, as discussed in Section 3.4. Using p independent replications of the simulation is a well known approach to obtain independent sequences of output data. Let $\left\{\{x_{j,i}\}_{i=1}^{n_j}\right\}_{j=1}^{p}$ denote the collected

observations. $x_{j,i}$ is the ith observation of the jth replication. n_j is an unbounded value which denotes how many observations are collected in the jth replication. Because we use synchronised replications only, we will assume that $\forall j : n_j = n$. Additionally, let us assume that $\forall j : F_{X_{j,i}}(x) = F_{X_i}(x)$ holds for a constant value of i, where $X_{j,i}$ is the random variable of the observation $x_{j,i}$. This means that the ith observation of all replications describes the same (possibly) transient measure. For example the ith observation could be the delay of the ith customer leaving a system, or it could be defined as the queue length at model time $i \cdot 100$ seconds if queues are observed at regular intervals of 100 seconds. These assumptions ensure that the data in the ith column is independent and identically distributed, i.e. Equation (2.4) and Equation (2.5) hold for $\{X_{j,i}\}_{j=1}^{p}$ for all i. These assumptions allow us to estimate a value of $F_{X_i}(x)$ by Equation (2.6). If the whole empirical CDF is of interest, it is advisable to base the estimation on a sorted random sample. Let $\{y_{j,i}\}_{j=1}^{p}$ be the ordered sequence of $\{x_{j,i}\}_{j=1}^{p}$. $\hat{F}_{X_i}(x)$ can be estimated by Equation (2.11) on basis of $\{y_{j,i}\}_{j=1}^{p}$.

4.1 Confidence Intervals in Rank and Probability Domain

A valid estimator for the location of the q-quantile at observation index i is given by

$$\hat{x}_{q,i} = y_{\lfloor pq+1 \rfloor, i} \tag{4.1}$$

(compare with Equation (2.14)). To simplify the notation, in the further text we will omit the dependence on i. The half width of a confidence interval of $\hat{x}_q$ can be described in two ways,

$$\text{either as} \quad \hat{x}_q \in x_q \pm \epsilon'_q, \quad \text{or as} \quad \hat{x}_q \in x_{q \pm \epsilon_q}.$$

ϵ'_q describes an interval in the range of the measure and ϵ_q describes an interval in the range of the probability (see [25-CK99]). Note, the interval $q \pm \epsilon_q$ should not

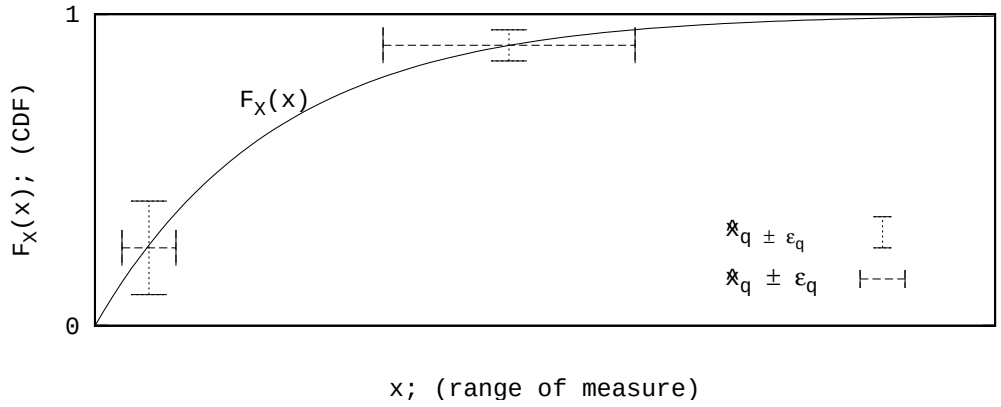


Figure 4.1: Confidence intervals for quantiles.

exceed the bounds 0 and 1. ϵ'_q and ϵ_q are dependent on each other: The interval $[q - \epsilon_q, q + \epsilon_q]$ in the probability domain has an associated interval $[x_q - \epsilon'_q, x_q + \epsilon'_q] = [y_{l,i}, y_{u,i}]$ given by the ranks l and u of order statistics. These ranks can be calculated by Equation (2.14). The calculation from ranks to probabilities can be done in a similar way by applying Equation (2.15). If one is decreased, e.g. ϵ'_q, the associated ϵ_q will decrease as well. However, in steep areas of $F_{X_i}(x)$ we expect ϵ'_q to be smaller (relatively) than ϵ_q. In flat areas of $F_{X_i}(x)$ we expect ϵ'_q to be bigger (relatively) than ϵ_q. This is demonstrated in Figure 4.1 with the example of an exponential distribution. Note, the difference in size of the confidence intervals between the steep and the less steep regions of the curve.

In general, ϵ'_q can be calculated from

$$\begin{aligned} \Pr\{y_{l,i} \leq x_q < y_{u,i}\} &= 1 - \alpha_{l,u} \\ &\geq \sum_{j=l}^{u-1} \binom{p}{j} q^j (1-q)^{p-j} \end{aligned} \tag{4.2}$$

by decreasing l and increasing u until the chosen confidence level $(1 - \alpha) \leq (1 - \alpha_{l,u})$ is reached (see Equation (2.13)). Note, we always calculate balanced confidence intervals as defined by Equation (2.20). l and u are both ranks in the ordered sample $\{y_{j,i}\}_{j=1}^{p}$ of the original observations $\{x_{j,i}\}_{j=1}^{p}$ and describe the

location of the lower and the upper border of the confidence interval. They should not exceed the borders 1 and p. Note that neither the value of the lower border $y_{l,i}$ nor the value of the upper border $y_{u,i}$ are involved in the calculation of the Equation (4.2).

In [25-CK99] it is demonstrated that ϵ_q can be chosen from the asymptotic Gaussian approximation

$$p \geq \frac{z^2_{1-\alpha/2} q(1-q)}{\epsilon_q^2}, \tag{4.3}$$

where $z_{1-\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution (compare also with [97-New98]). ϵ_q can be calculated for a given setting of p, q and $z_{1-\alpha/2}$. The confidence level α can be regarded as a constant parameter, and hence $z_{1-\alpha/2}$. q defines the quantile itself. p remains as the only important parameter. Note, ϵ_q does not depend on the collected observations.

Both, Equation (4.2) and Equation (4.3) do not depend on the output data itself. Therefore, both formula can be used to estimate the half width before the simulation experiment starts. From the point of view of mean value analysis, this is quite surprising as a confidence interval for an estimated mean value depends on the output data itself. However, Equation (4.2) and Equation (4.3) mainly depend on the number of replications p, because the confidence level $1-\alpha$ can be considered in both cases as a constant parameter. Therefore, p is the most important parameter in the methods described in subsequent sections.

To fully investigate the transient behaviour of a measure of interest an analysis of several quantiles over time is needed. As discussed above, the use of independent replications enables the estimation of $F_{X_i}(x)$ based on order statistics. However, is it really appropriate to use all order statistics, i.e. all of these $\frac{1}{p+1}, \ldots, \frac{j}{p+1}, \ldots, \frac{p}{p+1}$ quantiles to e.g. depict the transient behaviour? Because the confidence intervals of two adjacent quantiles at $\frac{j}{p+1}$ and $\frac{j+1}{p+1}$ overlap extensively (see Equation (2.13)), the estimates are highly correlated (see also Equation (2.18)). Thus, it is questionable to use both quantiles. To allow a clear de-

piction the quantiles should be chosen with non-overlapping confidence intervals. This suggests a method which determines a maximum number of quantiles with non-overlapping confidence intervals, for a given number of replications p, because the half width of the confidence interval of $\hat{x}_q$ depends on p.

4.2 Selection of Quantiles

The quantile estimates of two neighbouring order statistics $Y_{j,i}$ and $Y_{j+1,i}$ are correlated and the second quantile does not add much information to the estimation of the underlying distribution. On the other hand, if the distance between two consecutive quantiles is too large, the underlying distribution cannot be described appropriately. We try to avoid both extremes by selecting a set of quantiles as described in Section 2.3. The resulting set of quantiles have non-overlapping confidence intervals. Furthermore, these confidence intervals are balanced, as in Equation (2.20). This implies that the confidence intervals are not necessarily symmetric, due to the underlying binomial distribution of the order statistics. The confidence interval of the median is the only confidence interval that will be symmetric. All necessary calculation can be performed before the simulation experiment starts, and therefore, the run time of this method does not really matter. A linear or a binary search for more quantiles can be performed. The flowchart in Figure 2.1 shows a linear search.

The second method we investigate is based on Equation (4.3) and operates in the probability domain. As in the method described in Section 2.3, the starting point is the 0.5-quantile and the method searches for more quantiles in the directions above and below 0.5. In this case a linear or binary search is not needed, because the next quantile can be calculated directly using Equation (4.3) and the

following conditions:

$$q_k < 0.5: \qquad q_k - \epsilon_{q_k} = q_{k+1} + \epsilon_{q_{k+1}} \tag{4.4}$$

$$q_k > 0.5: \qquad q_k + \epsilon_{q_k} = q_{k+1} - \epsilon_{q_{k+1}} \tag{4.5}$$

q_k denotes the kth selected quantile. The first condition is valid for quantiles below the median and ensures that the upper bound of the confidence interval of the current quantile is equal to the lower bound of the previous confidence interval. The second condition is valid for quantiles above the median. It ensures that the lower bound of the new confidence interval is equal to the upper bound of the previous confidence interval. In the following we focus on the first condition, because the second condition can be treated analogously. We can assume that q_k is given or already calculated, because we initially choose $q_0 = 0.5$. ϵ_{q_k} can be calculated by Equation (4.3). Therefore, we can use the substitution $a_k = q_k - \epsilon_{q_k}$. Equation (4.4) can be transformed to:

$$a_k = q_{k+1} + z_{1-\alpha/2}\sqrt{\frac{q_{k+1}(1-q_{k+1})}{p}} \tag{4.6}$$

Eliminating the square root leads to

$$0 = q_{k+1}^2 b + q_{k+1} c_k + d_k \tag{4.7}$$

with $b = \frac{1}{z_{1-\alpha/2}^2} + \frac{1}{p}$, $c_k = -\frac{2a_k}{z_{1-\alpha/2}^2} - \frac{1}{p}$ and $d_k = \frac{a_k^2}{z_{1-\alpha/2}^2}$. Finally, the new q_{k+1}-quantile can be calculated by

$$q_{k+1} = \frac{-c_k - \sqrt{c_k^2 - 4bd_k}}{2b}. \tag{4.8}$$

Equation (4.8) is valid for quantiles below the median. An equation for quantiles above the median can be derived analogously. Furthermore, in the probability domain the location of the selected quantiles is symmetric with the median as their centre, except for errors due to rounding of non integer values. The selection of more quantiles is continued until the bounds of the probability domain

$[0, 1]$ are exceeded. With this approach the probability domain is filled with non-overlapping confidence intervals.

In [40-EMP05a] and [42-EMP06] we pointed out that there is not much difference in accuracy between Equation (4.2) and Equation (4.3). The selection approach based on Equation (4.3) seems less complex because no linear or binary search is needed. However, because Equation (4.2) enables us to calculate balanced confidence intervals (see Equation (2.20)), which are therefore asymmetric in general, we recommend to use the selection approach as described in Section 2.3. An example of a selected set of quantiles is given in Table 2.1 and Figure 2.1. All these calculations can be done as soon as the number p of replication is known, which is before the simulation experiment starts. Thus, the run time of the selection approach does not influence the run time of the simulation experiment.

4.3 Controlled Error

The confidence interval of each quantile is given by its lower bound $y_{l,i}$ and by its upper bound $y_{u,i}$ (see Equation (4.2)). u and l are ranks in the ordered sample. A confidence interval in the probability domain can be calculated on basis of l, u and Equation (2.15). It follows that this confidence interval is constant over i and does not depend on the output data itself. The error can be controlled by reducing the confidence interval's (relative) size to a specified threshold. By assuming a larger number of replications the sample size p is increased. With the approach of Section 2.3 the maximum confidence interval size in the probability domain can be determined. If the maximum size is larger than the specified threshold, the number of replications needs to be increased. In this way the error in the probability domain can be controlled.

The confidence interval's size in the domain of the measure itself is given by $y_{u,i} - y_{l,i}$. Here, the confidence interval's size is different for every i and can only

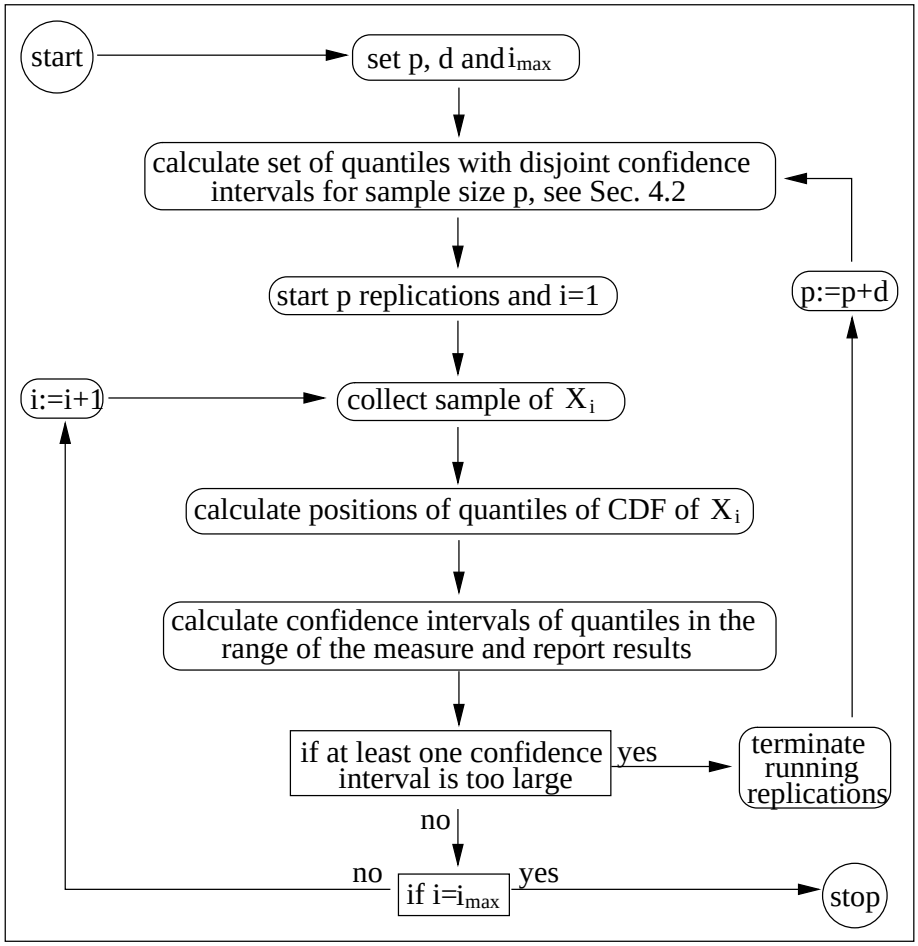


Figure 4.2: Flowchart for error control.

be calculated during the simulation run. Thus, this error cannot be controlled without knowing the output data. If a controlled error in the domain of the measure is wanted the error must be derived during simulation. In case of an unacceptable error more replications are needed to increase the sample size, then, l and u will be closer together and the difference $y_{u,i} - y_{l,i}$ will decrease. Increasing the number of replications can be done in two ways. Firstly, new replications could be added to the set of previous replications. Hereby, no output data gets lost, however, all output data needs to be stored. Storage requirement can be large. It depends on the specified threshold and the form of the underlying distribution function. In general, it cannot be guaranteed that available storage is sufficient.

Secondly, the number of replications can be increased by simply starting a larger number of replications and discarding previous replications and their output data. For this approach the data of the current observation index needs to be stored only. Discarding old results seems to be wasteful, however, this might be the only applicable solution. A flowchart of an implementation of this approach is given in Figure 4.2, where d defines how many replications are added in case of insufficient precision and $i_{\max}$ is the wanted horizon.

For practical reasons we recommend to control the error in the probability domain only. In addition the error in the range of the measure can be reported during the simulation experiment. If, for any reasons, the error in the range of the measure needs to be controlled, we recommend not to store simulation data. This assumes that a sufficiently large number of replications can be executed in parallel.

4.4 Examples

In the previous section we described how to select a number of quantiles. In this section we calculate quantiles for stochastic processes with known statistical properties to validate the results. This is followed by an investigation of the time evolution of quantiles of more complex models. These investigations show that the transient behaviour of quantiles is a very intuitive way to depict the transient behaviour of a given process. Our last example is an application in Peer-to-Peer file sharing systems. In all our simulations we used the random number generator described in [87-LSCK02], see Section 3.5. This generator allows the choice of many substreams, making it suitable for multiple independent replications.

4.4.1 Validation

The first experiments in this section are done to validate the estimated quantiles. We choose an *Autoregressive moving average* (ARMA) process with known

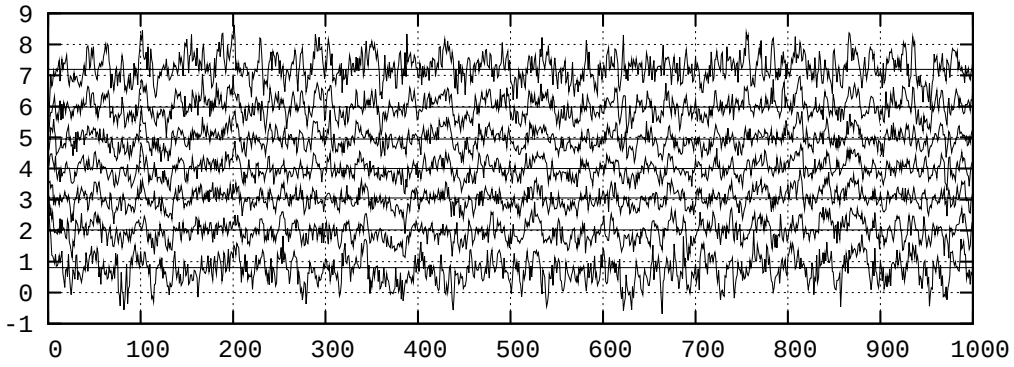


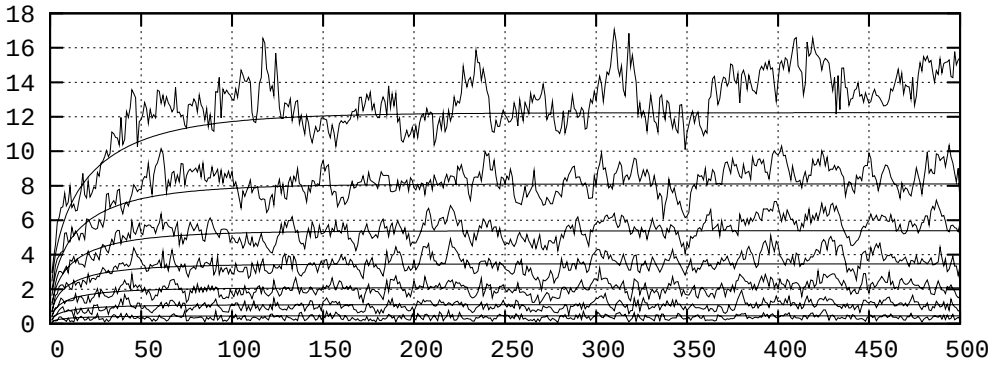
Figure 4.3: Time evolution of quantiles of a geometrical ARMA$(2, 2)$ process.

steady state behaviour, see Appendix A.2, as well as M/M/1 and M/E_2/1 queues. Their transient behaviour is analytically tractable. The calculation of expected values of the time evolution of quantiles of the response time is discussed in Appendix A.3 and Appendix A.4. Furthermore, the correlation of the output stream of the queueing models can be influenced by the traffic intensity ρ.

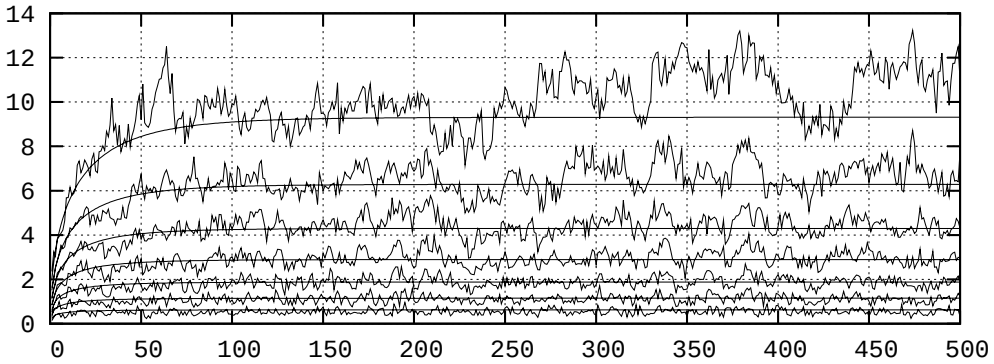
The simulation results are obtained by applying the estimation method using $p = 99$ replications and assuming the confidence level $1 - \alpha = 0.9$. The quantiles are selected with balanced and non-overlapping confidence intervals, as described in Section 2.3. We expect that the estimated quantiles follow the curve of the calculated quantiles. The curves of the calculated quantiles should be smooth because they are expected values not influenced by randomness. In contrast to this, the curve of the estimated quantiles are expected to show high frequency oscillations due to random errors. Note, that all quantile estimates and their confidence intervals are calculated for a specified confidence level $1 - \alpha$.

q	0.07	0.18	0.33	0.5	0.67	0.82	0.93
$F^{-1}_{X_\infty}(q)$	0.81	2.02	3.05	4	4.95	5.98	7.19

Table 4.1: Selected q-quantiles of $F_{X_\infty}(x) = \mathrm{N}\left(x; 4, \frac{117}{25}\right)$.



(a) M/M/1 queue



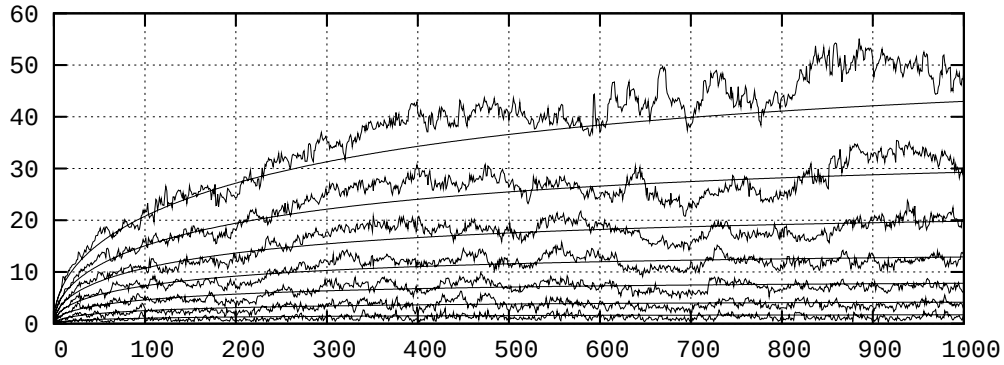
(b) M/E_2/1 queue

Figure 4.4: Time evolution of quantiles starting simulation with an empty queue and idle server. The traffic intensity is $\rho = 0.8$.

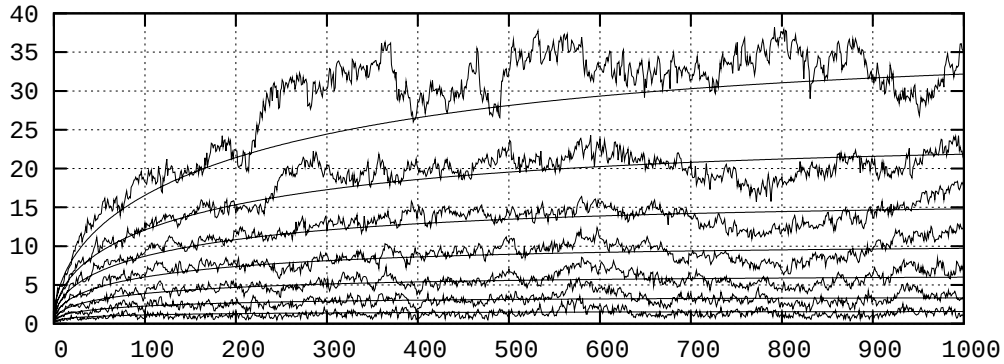
In Figure 4.3 the time evolution of quantiles of the geometrical ARMA$(2, 2)$ process

$$\Upsilon_i^{(2)} = 1 + \Psi_i + \sum_{k=1}^{2} \frac{1}{2^k} (\Upsilon_{i-k}^{(2)} + \Psi_{i-k}), \tag{4.9}$$

see Appendix A.2, is depicted. For the setting $p = 99$ and $1-\alpha = 0.9$ the quantiles listed in Table 4.1 are estimated. The true values are given by a normal distribution, because the CDF of $\Upsilon_i^{(2)}$ in steady state is given by $F_{\Upsilon_\infty^{(2)}}(x) = \mathrm{N}\left(x; 4, \frac{117}{25}\right)$, see Appendix A.2. The straight lines in Figure 4.3 show the true values of these quantiles for $i \to \infty$. The process is initialised by $\Upsilon_{-2}^{(2)} = \Upsilon_{-1}^{(2)} = \mathrm{E}\left[\Upsilon_\infty^{(2)}\right] = 4$. This initialisation leads to a very short transient phase, which is approximately



(a) M/M/1 queue



(b) M/E_2/1 queue

Figure 4.5: Time evolution of quantiles starting simulation with an empty queue and idle server. The traffic intensity is $\rho = 0.95$.

$1 \leq i \leq 20$. However, we depicted the evolution of the quantiles up to $i = 1000$. The estimated quantiles follow the straight lines, which represent the true values in steady state. Furthermore, we can see that the quantile estimates show high frequency oscillations, as expected. Every quantile $F^{-1}_{\Upsilon^{(2)}_i}(q)$ is estimated by exclusively using data collected at observation index i. Hereby, we can exclude any bias for the case $F^{-1}_{\Upsilon^{(2)}_i}(q) \neq F^{-1}_{\Upsilon^{(2)}_{i+\Delta}}(q)$, where $\Delta \neq 0$. However, this is also the reason why the high frequency oscillations of the curves of the estimated quantiles do not flatten out with increasing i. No matter which value of i is regarded, the intensity of the high frequency oscillations will remain constant, even in steady

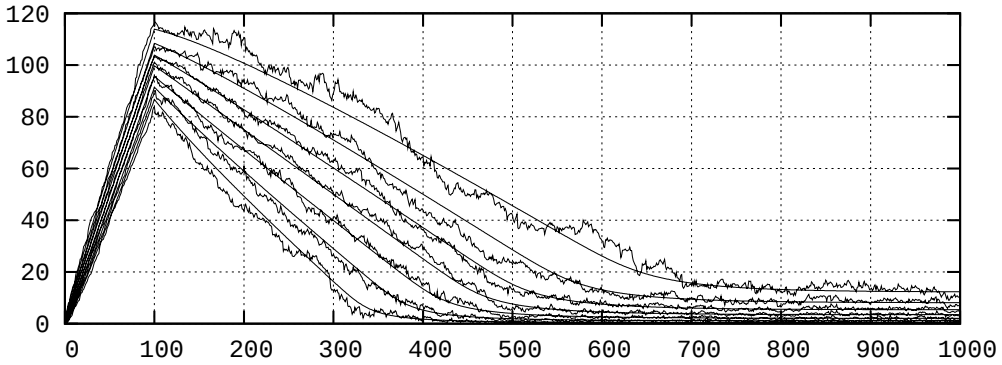


Figure 4.6: Time evolution of quantiles of the M/M/1 queue starting simulation idle but with 100 waiting customers. The traffic intensity is $\rho = 0.8$.

state.

In Figure 4.4 the results for the M/M/1 and the M/E_2/1 queue at a traffic intensity $\rho = 0.8$ are shown. At the beginning of the simulation the queues are empty and the servers are idle. This medium traffic load leads to a medium correlation of the output data (see [15-Ber79]). As one can see, the curves of the quantile estimates follow the curves of the expected values. This implies that the estimates are valid. The fluctuation due to randomness is also evident. The absolute variation of higher quantiles is higher than of lower quantiles due to the exponential character of the distribution.

Results for the M/M/1 and the M/E_2/1 queue at a traffic intensity of $\rho = 0.95$ are shown in Figure 4.5. In this case the high traffic load introduces higher correlation into the stream of output data. However, the results are similar to those for medium correlated output data. This shows that high correlation does not influence the quality of the estimates itself.

In our last experiment we choose again the M/M/1 queue at a traffic intensity $\rho = 0.8$, but here 100 customers are waiting in the queue when the simulation starts. This introduces a non monotonic behaviour of the curve of the estimated quantiles. In Figure 4.6 we can see that this more complex behaviour does not

influence the quality of the estimates, either. The estimated curves follow the expected curves.

These experiments provide validation of our method of estimation of quantiles over time. In these examples there is no evidence that either the form of the time dependent behaviour or the correlation of the output data influences the quality of the estimates. This suggests the estimation method is robust.

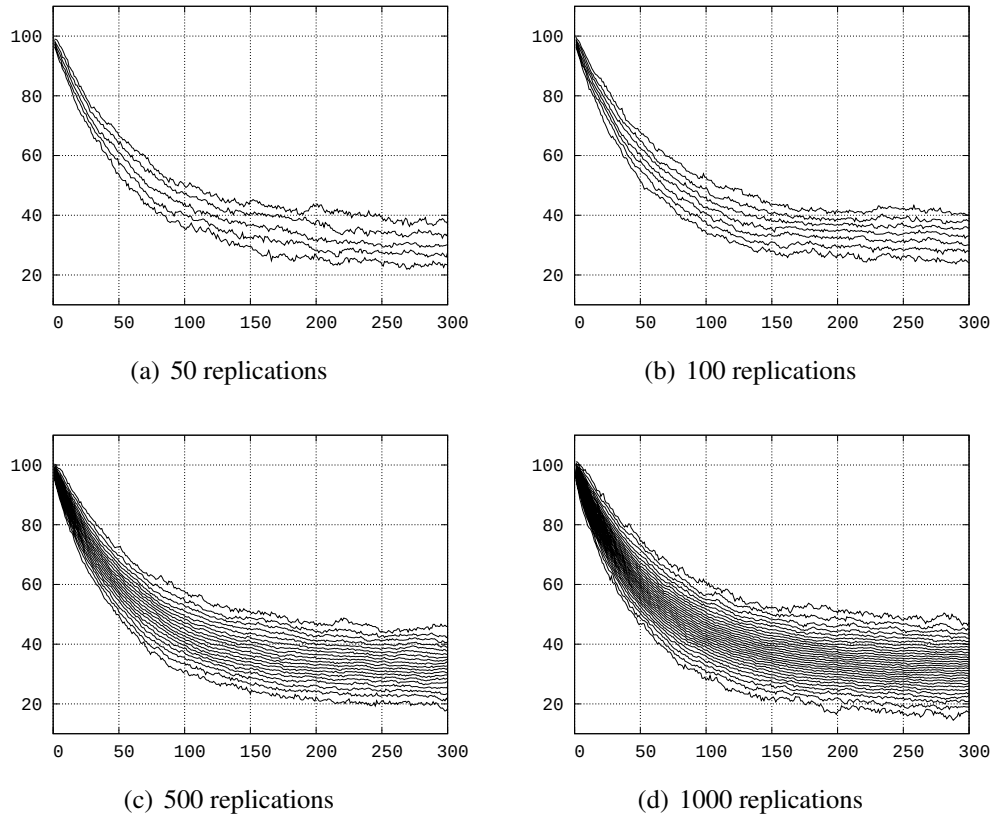
4.4.2 Common Types of Evolution over Time

The examples in this section are given to demonstrate the performance of the quantile estimation method over time on typical forms of time dependent behaviour. The underlying simulation models are artificial and quite simple, however, their behaviour can be regarded as representative of many other more complex simulation models. Note, a complex simulation model does not necessarily involve a complex behaviour of the output stream.

ARMA processes are commonly used in time series analyses. They are a class of stochastic processes with well known statistical properties. To show results of our method of transient quantile estimation we use an geometrical ARMA$(5,5)$ process, see Appendix A.2, which is defined by

$$\Upsilon_i^{(5)} = 1 + \Psi_i + \sum_{k=1}^{5} \frac{1}{2^k}(\Upsilon_{i-k}^{(5)} + \Psi_{i-k}), \tag{4.10}$$

with the starting condition $\Upsilon_{-5}^{(5)} = \Upsilon_{-4}^{(5)} = \Upsilon_{-3}^{(5)} = \Upsilon_{-2}^{(5)} = \Upsilon_{-1}^{(5)} = 100$. All Ψ_i are independent of each other and their distribution is the standard normal distribution. $\{\Psi_i\}_{i=1}^{\infty}$ is called an independent Gaussian white noise process in [67-Ham94]. Therefore, the process $\{\Upsilon_i^{(5)}\}_{i=1}^{\infty}$ is normally distributed for any i with a transient mean and variance. The expected value of this process for large i is $\mathrm{E}\left[\Upsilon_{\infty}^{(5)}\right] = 32$. This process is highly autocorrelated, because its current value depends on five previous values. The process is expected to converge from the initial value 100 to 32. The estimates of the transient quantiles are shown in Fig-



(a) 50 replications (b) 100 replications

(c) 500 replications (d) 1000 replications

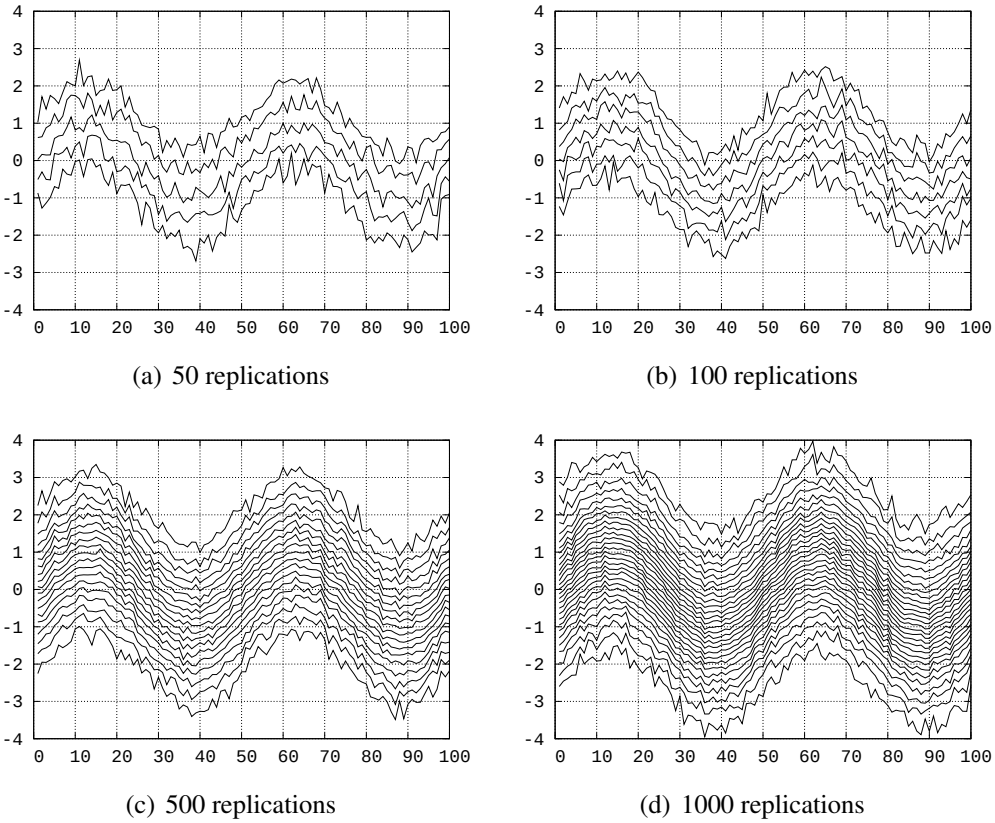
Figure 4.7: Several quantiles over time: geometrical ARMA$(5, 5)$ process.

ure 4.7. The simulation of the geometrical ARMA$(5, 5)$ process behaves exactly as expected. Additionally, we get an impression of the speed of the convergence, which is high in the beginning and increases with decreasing i. The underlying probability distribution of each $\Upsilon_i^{(5)}$ gets more evident, as more quantiles are used. Results of this example are published in [40-EMP05a].

The second examined stochastic process is *periodic* and is defined by

$$X_i = a \cdot \sin(\omega i) + \Psi_i \tag{4.11}$$

The cycle length of the sine oscillation is given by $T = \frac{2\pi}{\omega}$ with the amplitude a. We choose $T = 50$ and $a = 1$. Again $\{\Psi_i\}_{i=1}^{\infty}$ is an independent Gaussian white noise process. The estimates of quantiles are depicted in Figure 4.8. The periodic behaviour is visible for every depicted quantile. Again, the underlying probability



(a) 50 replications (b) 100 replications

(c) 500 replications (d) 1000 replications

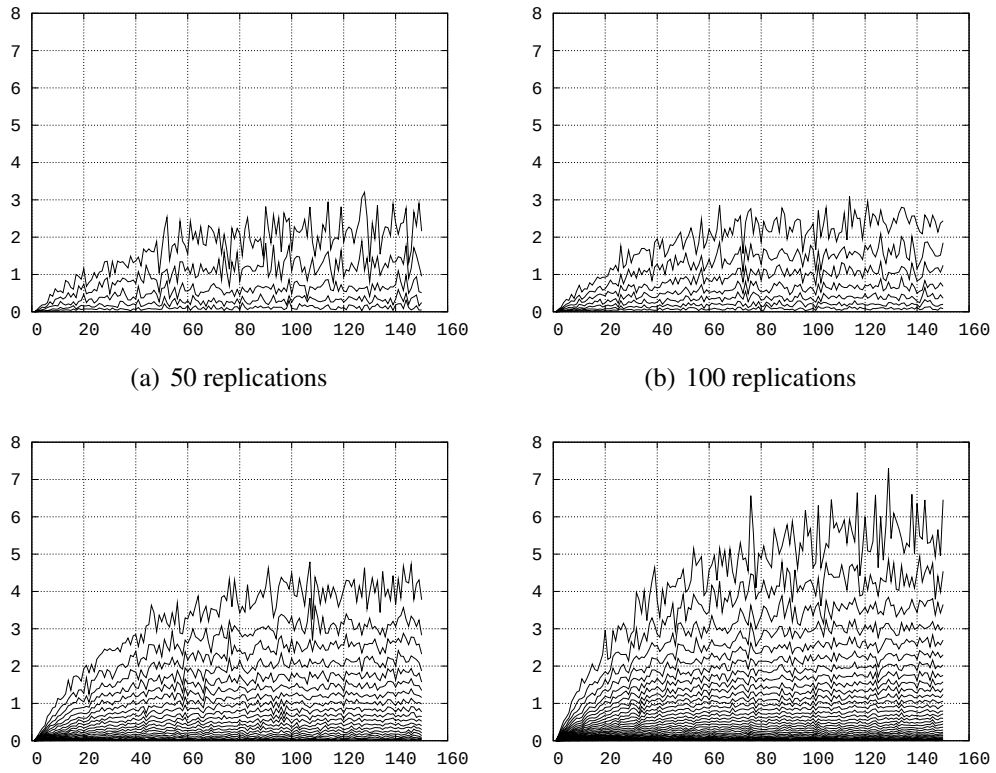
Figure 4.8: Several quantiles over time: periodic process.

distribution of each X_i gets more evident, the more quantiles are depicted. Results of this example are published in [40-EMP05a].

In the previous example we estimated quantiles of normally distributed processes. In this example we chose a process which is governed by an *exponential distribution*. It is defined by

$$X_i = \Psi'_i \cdot b(1 - e^{(i\frac{ln(0.05)}{l})}). \tag{4.12}$$

The process $\{\Psi'_i\}_{i=1}^{\infty}$ is similar to the independent Gaussian white noise process, but its distribution is exponential with mean one, i.e. $F_{\Psi'_i}(x) = \text{Exp}(x; 1)$. The parameter b stretches the distribution. The part in brackets of the formula causes the process to slowly converge towards its marginal distribution. This is depicted in



(a) 50 replications

(b) 100 replications

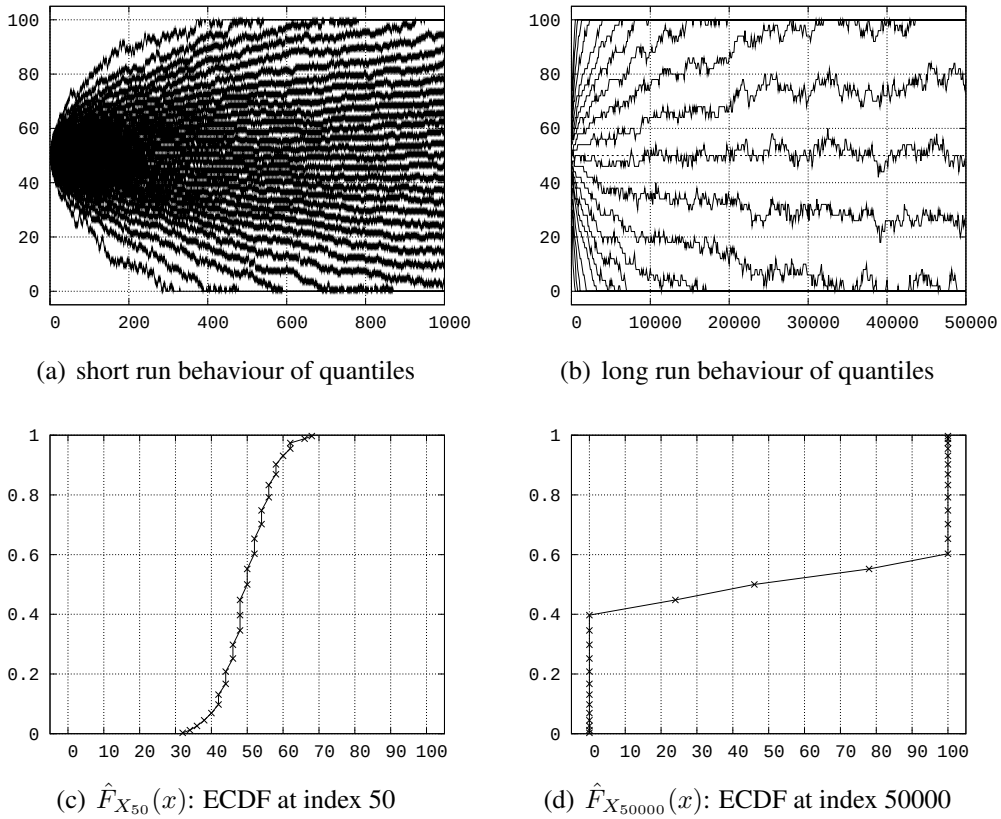
(c) 500 replications

(d) 1000 replications

Figure 4.9: Several quantiles over time: exponential process.

Figure 4.9. Both the convergence and the exponential character of the distribution are clearly apparent. Results of this example are published in [40-EMP05a].

In general, the quantiles of areas with lower probability density seem to fluctuate more (absolute not relative) than the ones of areas with high probability density. In Figure 4.7 and Figure 4.8 this can be observed when comparing the bounds 0 and 1 with the centre (around 0.5) of the distribution. Because the distribution in Figure 4.9 is not symmetrical, the quantiles at bound 1 fluctuate more than the ones at bound 0. These examples show, that our approach of depicting quantiles is suitable for both symmetrical and asymmetrical distributions, as well as for converging and non converging processes. In [40-EMP05a] it is recommended to use at least 50 independent replications to ensure a set of at least 5



(a) short run behaviour of quantiles

(b) long run behaviour of quantiles

(c) $\hat{F}_{X_{50}}(x)$: ECDF at index 50

(d) $\hat{F}_{X_{50000}}(x)$: ECDF at index 50000

Figure 4.10: Quantiles and ECDFs of a bounded random walk.

different quantiles with $\alpha = 0.1$.

4.4.3 More Complex Examples

In this section we will apply the quantile estimator on the simulation output of models with more complex behaviour. The structures of the models itself are still quite simple. However, being unaware of their complex evolution over time would lead to errors in common simulation output analysis, especially in mean value analysis.

The first process is based on a *random walk* X'_i, which is defined by

$$X'_i = \begin{cases} X'_{i-1} + 1, & \text{with probability } 0.5, \\ X'_{i-1} - 1, & \text{with probability } 0.5, \end{cases} \tag{4.13}$$

with the initial state $X'_0 = 50$. The process X'_i can take any value between $-\infty$ and $+\infty$. The final process X_i is bounded, so that its range is the interval $[0, 100]$:

$$X_i = \begin{cases} 0, & \text{if } X'_i < 0, \\ X'_i, & \text{if } 0 \leq X'_i \leq 100, \\ 100, & \text{if } X'_i > 100. \end{cases} \tag{4.14}$$

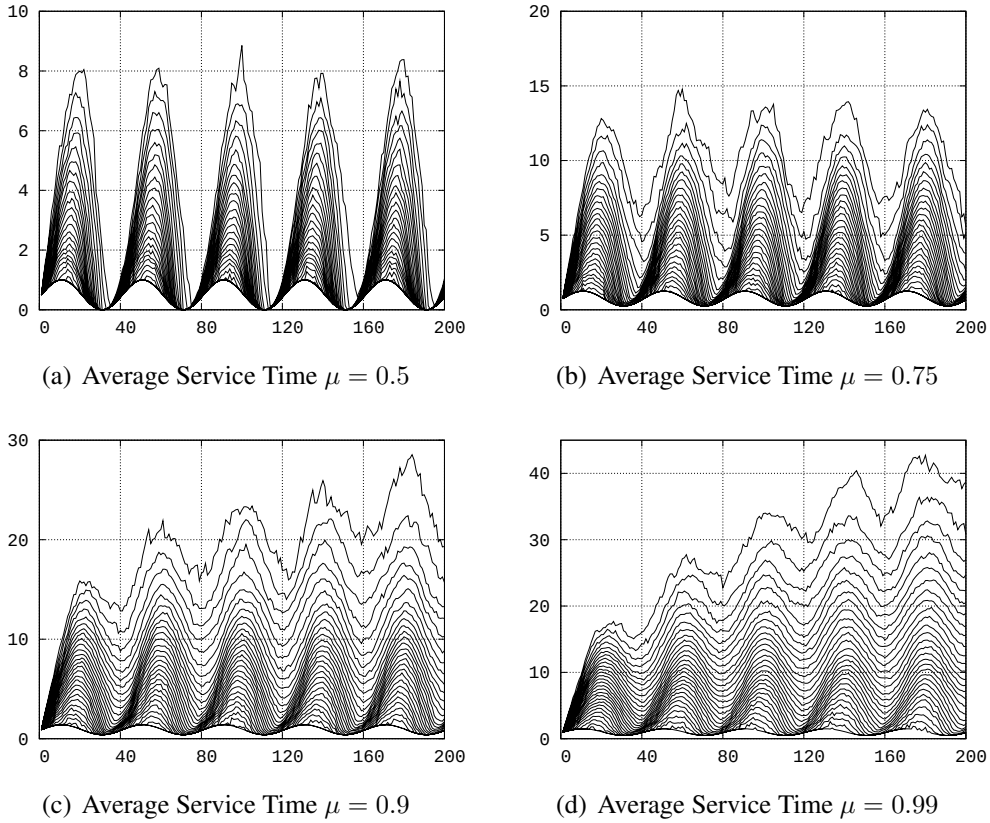
A similar process was used in [12-BB99]. Because X_i is bounded a marginal distribution for $i = \infty$ exists.

The peculiarity of this process is that the expected value $\mathrm{E}[X_i] = 50$ is constant over i, whereas all quantiles other than the median are not constant and converge to the thresholds 0 and 100, see Figure 4.10(a) and Figure 4.10(b). $F_{X_i}(x)$ is very steep around $x = 50$ for small i, see Figure 4.10(c). After a long simulation time the shape of $F_{X_i}(x)$ is completely different. For large i it is very flat around $x = 50$, see Figure 4.10(d). However, the expected value $\mathrm{E}[X_i]$ is constant for all i. Analysis of mean values only would show a constant behaviour, even though quantiles of this process are transient and the cumulative distribution is slowly converging to its marginal distribution. Results of this example are published in [40-EMP05a] and [42-EMP06].

A *periodic behaviour* can be introduced into a queueing model in two ways. On the one hand, the system arrivals could be governed by an oscillating function. On the other hand, the service process could be influenced by an oscillating function. In this example we choose a single server system with an unbounded queue. The interarrival process is a Poisson process. The service process is deterministic and periodic. We denote this queueing process as $M/D_{\textit{periodic}}/1$. The service time μ_i of the ith customer is defined by:

$$\mu_i = a \cdot \sin(\omega i) + \mu \tag{4.15}$$

The average service time μ is a positive value. a is the amplitude, with $0 \leq a \leq \mu$, to avoid negative values of the service time μ_i. The cycle length $T = \frac{2\pi}{\omega}$ of the



(a) Average Service Time $\mu = 0.5$

(b) Average Service Time $\mu = 0.75$

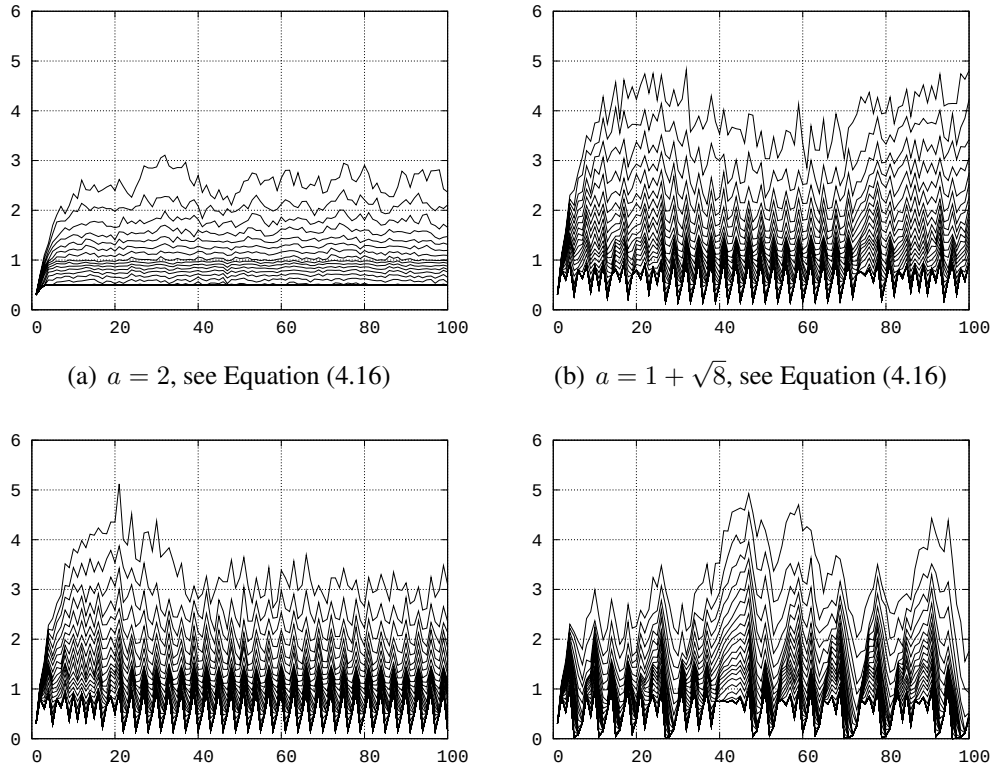
(c) Average Service Time $\mu = 0.9$

(d) Average Service Time $\mu = 0.99$

Figure 4.11: Quantiles of the response time of the $M/D_{periodic}/1$ system.

sine oscillation is also a positive value.

In our experiments we choose $\mu = \{0.5, 0.75, 0.9, 0.99\}$, $a = 0.5$, $T = 40$ and the average interarrival time is 1.0. We observed the response time, i.e. the time spend in queue plus the time spend in service, of consecutive customers. The results are depicted in Figure 4.11. Note that the plots have different vertical scales. The periodic influence is clearly evident. The peaks of higher quantiles are shifted by about $T/4$, whereas the peaks of lower quantiles stay close to the original periodic behaviour. Higher quantiles describe long queue length. Therefore, this suggests that a long queue damps the effect of the periodic behaviour. The peaks become higher and wider for an increasing μ so that they grow together (compare Figure 4.11(a) and Figure 4.11(d)). Results of this example are published in



(a) $a = 2$, see Equation (4.16)

(b) $a = 1 + \sqrt{8}$, see Equation (4.16)

(c) $a = 1 + \sqrt{8} + 0.01$, see Equation (4.16)

(d) $a = 4$, see Equation (4.16)

Figure 4.12: Quantiles of the response time of the $M/D_{logistic}/1$ system.

[42-EMP06].

Chaotic systems are nonlinear, aperiodic and depend heavily on initial conditions. Usually they have a control parameter, which can cause the chaos to appear or disappear. The logistic equation

$$\mu_i = a\mu_{i-1}(1 - \mu_{i-1}) \tag{4.16}$$

shows chaotic behaviour if the initial state μ_0 is not a fixed point of Equation (4.16). This would lead to a constant μ_i. a is a positive constant with $0 < a \leq 4$. For some settings of a the process μ_i converges to one value. For other settings of a it jumps between a certain number of values after an initial phase. And for some settings of a the process μ_i shows no pattern at all. Small changes of a can lead to

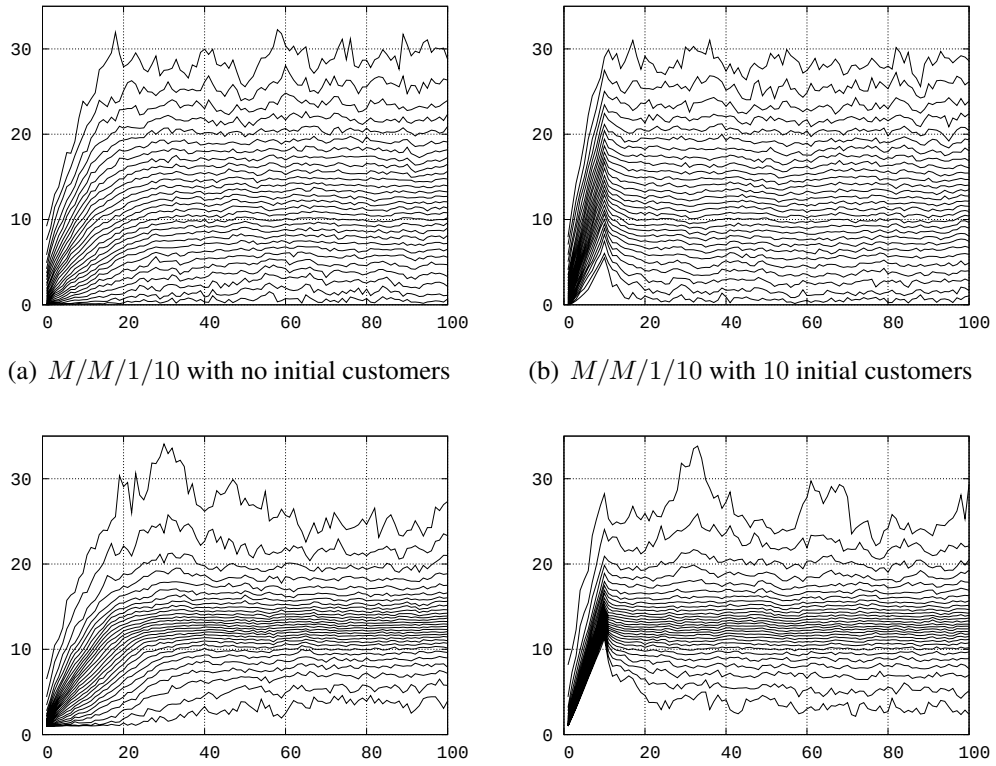
completely different behaviour of μ_i. For a detailed discussion on Equation (4.16) see [125-Spr03]. If the logistic equation is implicitly hidden in a model, it is very hard to get an insight into its behaviour by analytical methods. We choose the logistic equation to define the service time μ_i of the ith customer in a single server system. This explicitly introduces a chaotic behaviour and we incorporate it in the queueing model $M/D_{logistic}/1$. A process of this kind is analytically tractable only if the exact value of a is known.

In our experiments we observed the response time of consecutive customers. The average interarrival time of the Poisson process is 1.0. We set $a = \{2, 1 + \sqrt{8}, 1 + \sqrt{8} + 0.01, 4\}$ and $\mu_0 = 0.3$. For $a = 2$ (see Figure 4.12(a)) the queueing model shows a short warm up period. After this, μ_i is constant, and therefore, the estimated transient quantiles seem to be stable. The point $a = 1 + \sqrt{8}$ is the onset of a window, in which μ_i jumps between three values. This is depicted in Figure 4.12(c). Figure 4.12(b) does not show this behaviour, even though the value of a is very similar. For $a = 4$ the depiction of the quantiles does not show any pattern. Furthermore, the time evolution of higher quantiles is not always exactly comparable to those of lower quantiles. For example between the 40th and the 45th customer in Figure 4.12(d) the lowest quantile is on a constant high level but higher quantiles are increasing. Results of this example are published in [42-EMP06].

Next we compare two common queueing models with *bounded queue length*, the $M/M/1/10$ queue and a $M/P/1/10$ queue. In the second queue the service process $X^{(P)}$ is governed by the Pareto distribution

$$F_{X^{(P)}}(x) = 1 - x^{-\alpha}, \text{ where } x > 0. \tag{4.17}$$

To ensure that the first and the second moment of the Pareto distribution exist, we choose $\alpha = 3$. Therefore, the mean $\mathrm{E}\left[X^{(P)}\right] = 1.5$ and the $\mathrm{Var}\left[X^{(P)}\right] = 0.75$ are finite. To obtain comparable results, we choose the service process $X^{(M)}$ of the $M/M/1/10$ queue with the same expected value $\mathrm{E}\left[X^{(M)}\right] = 1.5$. The



(a) $M/M/1/10$ with no initial customers

(b) $M/M/1/10$ with 10 initial customers

(c) $M/P/1/10$ with no initial customers

(d) $M/P/1/10$ with 10 initial customers

Figure 4.13: Quantiles of the response time of the $M/M/1/10$ system in comparison with the $M/P/1/10$ system.

variance is in this case $\mathrm{Var}\left[X^{(M)}\right] = 2.25$. In both queueing models the average interarrival time is 1.0 and the maximum permitted queue length is nine customers plus one customer in service. Customers which arrive at a completely filled queue are rejected. Both queues are stable because their queue length is bounded.

We observed the response times in the two models for accepted customers. The results of our transient quantile estimation are shown in Figure 4.13. The quantiles converge to their steady state values. By comparing Figure 4.13(a) and Figure 4.13(c) it becomes obvious, that the probability distribution of the $M/P/1/10$ model is more centred around its expected value than the steady state distribution of the $M/M/1/10$ model. This is due to its smaller variance:

$\mathrm{Var}\left[X^{(P)}\right] < \mathrm{Var}\left[X^{(M)}\right]$. The highest quantile of the $M/P/1/10$ model fluctuates more than the highest quantile of the $M/M/1/10$ model. Due to our choice of α, higher moments than $\alpha > 3$ of the Pareto distribution do not exist, so this may cause higher fluctuation (absolute) of higher quantiles compared to lower quantiles. In an additional experiment we started the replications with a completely filled queue. Theses results are plotted in Figure 4.13(b) and Figure 4.13(d). The 10 initial customers engender a non-monotonic convergence of the quantiles. For more information about quantile estimation of a $M/P/1$ model see [48-FMG+01]. Results of this example are published in [42-EMP06].

4.4.4 File Popularity in Peer-to-Peer Networks

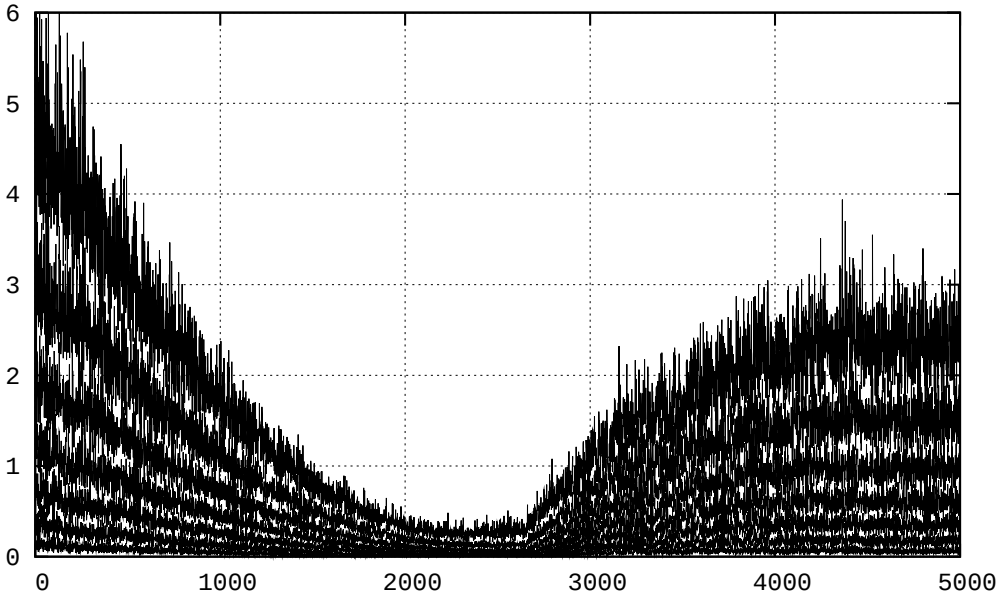
In the last years Internet technologies and infrastructures have experienced deep advancements and evolutions to meet the increasing user requirements and to support new application requests. It is well known that the major part of traffic carried by Internet is cause by file-sharing applications. It is not caused by the classical Internet data applications, like e-mail or HTML clients. In [17-BEPS07] a standard simulator is improved by using a more detailed query generation process. This process introduces dynamics of file popularity.

The file popularity function is given by

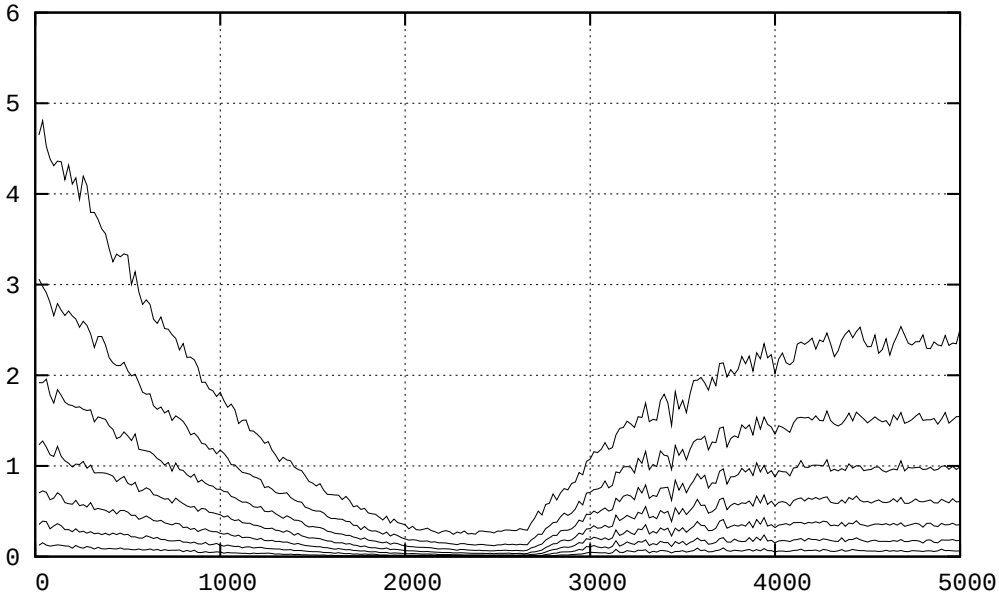
$$\psi(t) = \begin{cases} (e^{\frac{2t}{T-1}})/(e-1) & \text{if} \quad 0 \leq t < \frac{T}{2}, \\ \frac{T}{2t} & \text{if} \quad \frac{T}{2} \leq t < T, \\ 0.5 & \text{if} \quad T \leq t, \end{cases} \tag{4.18}$$

where e.g. $T = 365$ days. This time dependent popularity introduces a dynamic behaviour into the sequence $\{A_i\}_{i=0}^{\infty}$ of query interarrival times. In general $F_{A_i}(x) \neq F_{A_j}(x)$ can be assumed if $i \neq j$ and $i, j < T$. In this application exponentially distributed interarrival times are chosen, so that

$$F_{A_i}(x) = \mathrm{Exp}(x; \lambda/[1-\psi(t)]). \tag{4.19}$$



(a) quantile estimates



(b) smoothed quantile estimates

Figure 4.14: Time evolution of quantiles of $F_{A_i}(x)$ in seconds$\cdot 10^4$ (ordinate) of the ith query (abscissa).

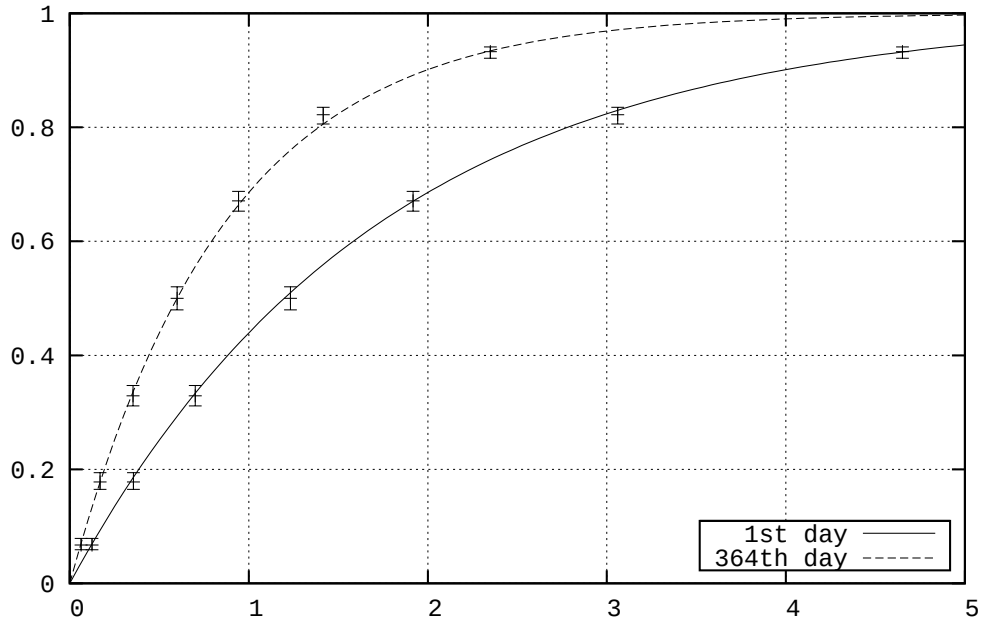


Figure 4.15: Cumulative probability (ordinate) over query interarrival time in seconds$\cdot 10^4$ (abscissa): Interval estimates of quantiles with $\psi(1)$ and $\psi(364)$ checked against the expected CDF.

For small i a long interarrival time is expected, i.e. A_i should show relatively large values. With increasing i up to the maximum of the file's popularity the interarrival time A_i should decrease to a low level, implying that a relatively large number of queries take place. After this period A_i should grow again with increasing values of i because the popularity is shrinking. The evolution over time of the quantile estimates are shown in Figure 4.14 and Figure 4.15 and are used to verify that the query interarrival time A_i follows the exponential distribution.

In Figure 4.14 we depicted quantiles of $F_{A_i}(x)$ for increasing i. In this case we considered a network with 2500 peers, 25 ultra peers and an initial number of 50 files (up to 600 considering replicas). Using $\alpha = 0.1$ and $p = 99$ independent replications 7 quantiles could be selected with non-overlapping confidence intervals.

Figure 4.14(a) shows the original quantile estimates, whereas a smoothed

curve for every quantile is shown in Figure 4.14(b). The original estimates are smoothed by averaging 20 consecutive values. As we can see, the behaviour is as expected. The curve of the quantile estimates of A_i start and finish on a relatively high level indicating a low popularity. In between they are on a low level caused by higher popularity. The big variance in the behaviour of each single quantile is because interarrival times are independent of each other.

We verified the exponential distribution of A_i at the first and the 364th day of the simulation with $\psi(1) = 0$ and $\psi(364) = 0.5$ (see Figure 4.15). The smoothed quantiles and their confidence intervals are checked against $F_{A_i}(x)$. As one can see, the estimates are as expected. For more details of this example see [17-BEPS07].

4.5 Limits and Conclusion

We have developed a method of quantile estimation so that it can deal with the evolution of quantiles over time. The set of quantiles is chosen automatically with regard to the sample size, given by the number of replications. All quantiles are estimated with confidence intervals, which are balanced and disjoint. This assures that the level of correlation of quantile estimates is not too high, even though they are taken from the same random sample. The exact degree of correlation depends on the confidence level $1 - \alpha$ and the form of the underlying probability function because Equation (2.18) depends on order statistics of the observed random sample.

Experiments with selected models in the previous section indicate that the estimation method is robust. It can deal with highly correlated output data, as well as with non monotone, periodic or even chaotic behaviour. The quality of the estimates is not influenced by these kind of characteristics of the simulation output data.

Analysis of the time evolution of the quantiles of a measure provides a deep

insight into the behaviour of the model itself. This can be used to understand the dynamics of the model as well as to verify its behaviour. In modern simulation tools an animation of the dynamics is common to show the dynamics of the created model during simulation. Our estimation method for the time evolution of quantiles supports this animation from point of view of output data analysis. It does even more: An animation can only show one possible behaviour of the model but the time evolution of quantiles shows all possible forms of behaviour of a measure in connection with its probability.

An application in the area of Peer-to-Peer file sharing systems is demonstrated. In this example interarrival times are influenced by a time dependent function. The simulator is validated by applying the quantile estimation over time. The curves of the quantile estimates behave as expected. This indicates that the simulator works correct.

The number of estimated quantiles depends on the size of the random sample, thus, on the number of replications. To obtain more quantile estimates the number of replications must be increased. Hereby, output data of old replications could be stored and output data of new replications could be added. On the other hand, if storage requirements are too high, output data of old replications could simply be discarded and a higher number of new replications could be started. The decision of which approach should be applied depends on the hardware restrictions of the given computer system.

Chapter 5

Initial Transient Phase

In every simulation experiment the initial state of a given model has to be set. Assuming that the main focus of analysis is the long run behaviour, a good initial state would be a typical state of the long run behaviour. However, because the long run behaviour of the model is unknown this kind of initialisation is not possible. Thus, we have to take into account that the initial state is atypical and that the simulation needs some time to recover from the impact of the initial state. Note, depending on the model and the initial state the simulation might never recover from the initial state. To avoid a heavily biased final estimate a common approach is to split the observed output data into two parts separated by the truncation point l. Parts of the discussion and results of this chapter are published in [41-EMP05b], [38-Eic06] and [43-EMP07a].

The convergence towards the long run behaviour can be very different and is depending on the simulated model. Some of the methods for the detection of l that are known so far assume special kinds of convergence and violate other kinds of convergence. Some methods are only valid for e.g. mean value analysis. Here, the aim is to detect a value of l, which is valid for many kinds of convergence and for the estimation of the mean, variance, quantiles or other measures.

For automated simulation analysis it is very important that l can be detected for a wide range of output processes. As mentioned, a steady mean value is only

a necessary condition for steady state. Furthermore, the convergence to the steady state behaviour is not necessarily monotonic. There are also situations where l cannot be estimated because it does not exist. This could be cause by an unstable simulation model cause by e.g. a too high load. An output process with periodic behaviour is also an example, where no l can be determined. A method for detecting l should be able to deal with all of this situations. Critical surveys can be found for example in [54-GAM78], [99-Paw90], [37-Eic02], [90-LH02] and [91-MI04]

Recently a statistical process control approach was discussed by [113-Rob07] and [112-Rob02] in the context of simulation output analysis to distinguish between the initial transient and steady state. Like the proposed methods of this thesis, this approach applies parallel replications and homogeneity tests (see Section 5.3). As we will discuss in this chapter, the use of replications and homogeneity tests assist the detection of steady state in terms of the underlying probability distribution. In contrast to this, in [113-Rob07] the advantage of obtaining a random sample at each observation index by collecting output data of independent replications is used to calculate a secondary process consisting of averaged observations. If the number of replications is large, normality of the averaged observations can be assumed, which assists the detection of a steady state mean. However, by averaging observations information about their original probability distribution gets lost. This is the reason why the approach of [113-Rob07] can only be used in mean value analysis and is not able to detect steady state in terms of the underlying probability distribution. In [114-RLQ05] the method of [112-Rob02] is applied to a secondary output process, which is the exponentially weighted moving average obtained by smoothing the original output process. On basis of the variance of the underlying data upper and lower control limits are calculated as a function of the observation index. The truncation point is determined on basis of a control measure being in relative position to selected control limits. Optimal settings of

various parameters, which are needed for the calculation of the control limits, are depending on the output process itself. Thus, the use of this method in automated simulation analysis is limited and a sequential execution is not discussed.

Another approach to reduce the initial bias is discussed in [9-AG06], which is related to [58-GI87], compare also with [59-GH91a], [60-GH91b], [61-GH92a] and [62-GH92b]. The described approach aims at estimating a truncation point after which random variables can be assumed to be identically distributed. Because it is based on equality in distribution, this approach is superior to other truncation point detection rules, which demand e.g. a constant mean only. It is applicable for one single simulation run as well as for multiple replications. The need for a fully automated approach by avoiding unspecified parameters is underlined. However, algorithmic properties of this approach, such as time complexity, storage requirements and sequential execution, are not discussed. This approach is discussed from the point of view of steady state analysis of mean values only.

A method to detect steady state is introduced by Welch, see [137-Wel83]. Welch remarked that this method checks the necessary condition of a steady mean only. Despite of this, the method can probably be regarded as the most common method for detecting steady state. The original output data is smoothed by averaging the values of a moving window. In this way the analyst can distinguish between random and systematic errors. This method is not acceptable in our case, because the determination of l depends on a visual inspection of smoothed data, no test is used to assure confidence in statistical sense. Furthermore, Welch's method will erroneously detect a value for l in case of an periodic output process, even though a valid l does not exist, see [14-BE03].

This situation of detecting the steady state is discussed in more detail in the next sections. A definition for l is given in Section 5.1. In Section 5.2 we give an example of how different initial states influence the output process. Section 5.3 to Section 5.5 discuss a new class of methods to detect the truncation point which

are based on comparing the probability distribution of the output process at different observation indexes. The results of this chapter are tested and concluded in Section 5.6 and Section 5.7.

5.1 Concept of Steady State Phase

The output stream of a simulation run is a stochastic process $\{X_i\}_{i\in\mathbb{N}}$. Due to the problem of initialisation X_i, as $i \to 0$, might not be representative of the system's usual behaviour. In many simulation studies the target is to analyse the system's behaviour in the long run, i.e. to analyse X_i, as $i \to \infty$. Because of obvious practical reasons X_∞ is not directly accessible. Thus, the concept of the *steady state phase* is introduced in simulation output analysis. During the steady state phase, $i \geq l$, the output data is representative of the system's behaviour and is (approximately) not influenced by the initial state. l is called the truncation point. Steady state in terms of the probability distribution is given if

$$\forall(i \geq l_F, \Delta \geq 0, x) : F_{X_i}(x) \simeq F_{X_{i+\Delta}}(x). \tag{5.1}$$

By "$\simeq$" we denote closeness of distributions, for example in the Kolmogorov sense:

$$\sup_{-\infty<x<\infty} |F_{X_i}(x) - F_{X_{i+\Delta}}(x)|. \tag{5.2}$$

Other interpretations of the operator "$\simeq$" are possible, for example in the Anderson-Darling sense. See Section 5.3 for details of the implementation of this operator. We use l_F instead of only l to explicitly point out that the truncation point is determined by inspecting $F_{X_i}(x)$. It is known that different performance measures, in particular different moments, converge to steady state at different rate; see Section 5.2 for examples. Often the convergence of $F_{X_i}(x)$ towards $F_{X_\infty}(x)$ is slow and cannot be completely attained by a finite value of i, therefore, approximate equality instead of strict equality is demanded. In this sense we call a process

stable in terms of the probability distribution if Equation (5.1) can be fulfilled by a finite value of l_F. We define the steady state phase in terms of the mean by

$$\forall(i \geq l_E, \Delta \geq 0) : \mathrm{E}\left[X_i\right] \approx \mathrm{E}\left[X_{i+\Delta}\right]. \tag{5.3}$$

Equation (5.3) can be used if the only target of the simulation is to estimate $\mathrm{E}\left[X_\infty\right]$. Equivalently, we define the steady state phase in terms of the variance by

$$\forall(i \geq l_V, \Delta \geq 0) : \mathrm{Var}\left[X_i\right] \approx \mathrm{Var}\left[X_{i+\Delta}\right]. \tag{5.4}$$

We will use l_F, l_E and l_V to explicitly mention the definition of the truncation point. l will be used if no definition is apparent or preferred. More kinds of truncation points are possible, e.g. for the 0.95-quantile of the distribution of X_i the truncation point $l_{0.95Q}$ could be defined analogously to Equation (5.3) and Equation (5.4).

Constant first and second moments are necessary conditions for Equation (5.1). Thus, steady state in terms of the probability distribution implies that the mean and the variance are in their steady state, i.e. $l_E \leq l_F$ and $l_V \leq l_F$. However, whether $l_E \leq l_V$ or $l_E \geq l_V$ holds depends on the output process. Both situations are possible, see Section 5.6 for examples. Note that it is possible to find processes for which Equation (5.3), and/or Equation (5.4), hold but not Equation (5.1). In this situation $\mathrm{E}\left[X_\infty\right]$ can be estimated even though $F_{X_\infty}(x)$ does not exist. Whether this makes sense or not has to be decided by the analyst. The counterpart of the steady state phase is the *transient phase* with $i < l$. During the transient phase Equation (5.1) does not hold.

In analysis of stochastic processes their *stationarity* is an important property and is discussed e.g. in [98-Pap84], [88-LG89] and [131-Tri02]. A stochastic process $\{X(t)\}_{t\in T}$, not necessarily representing simulation output data, is stationary (in the strict sense) if its statistics are not affected by a shift in the time origin. This means that two processes $\{X(t)\}_{t\in T}$ and $\{X(t+\Delta)\}_{(t+\Delta)\in T}$ have the same

statistics for any Δ. The joint distribution of any set of samples of a stationary process does not depend on the placement of the time origin:

$$F_{X(t_1),\ldots,X(t_j)}(x_1,\ldots,x_j) = F_{X(t_1+\Delta),\ldots,X(t_j+\Delta)}(x_1,\ldots,x_j), \qquad (5.5)$$

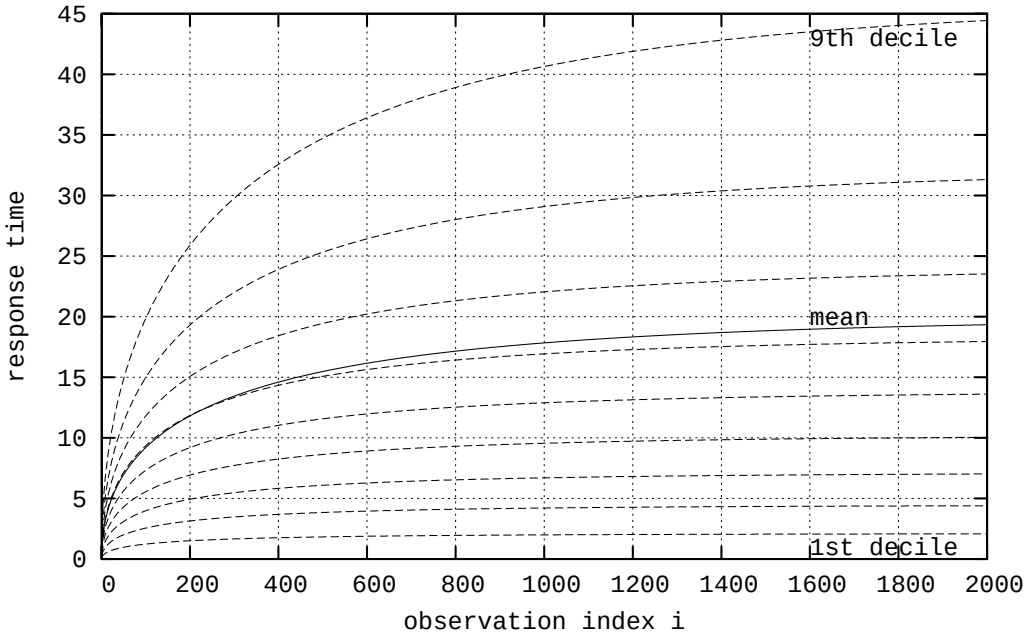
for all time shifts Δ, all j and all choices of sample times t_1, ..., t_j. If Equation (5.5) is true, not for any j, but only for $j \leq k$ the process $\{X(t)\}_{t\in T}$ is stationary of order k. Therefore, the simulation output process $\{X_i\}_{i\geq l_F}$ can be assumed to be stationary of first order, if Equation (5.1) can be fulfilled.

5.2 Convergence of M/M/1 Queues

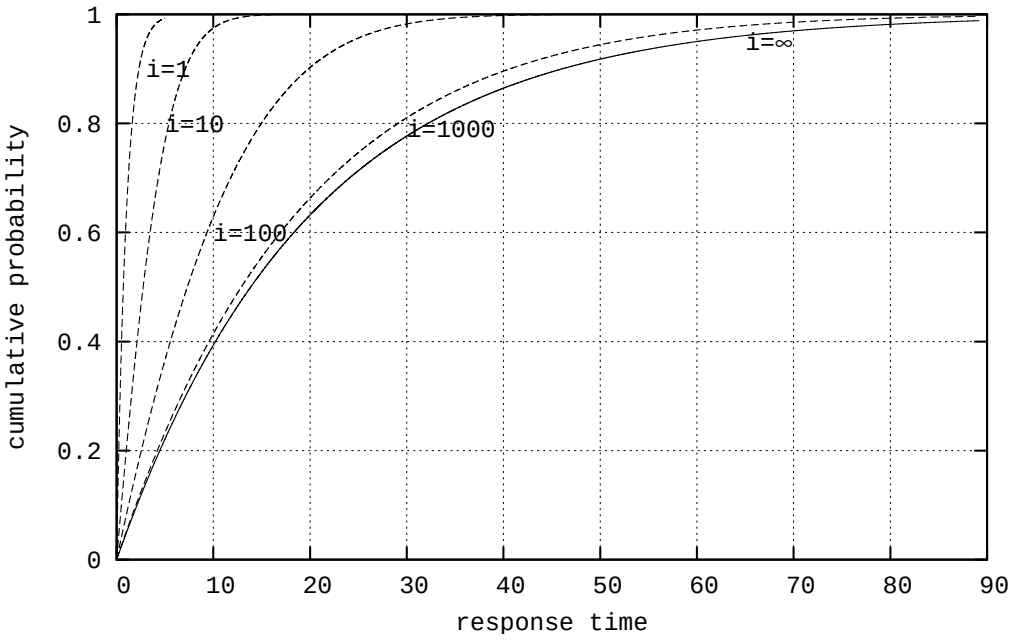
In this section we study the convergence of $F_{X_i}(x)$ towards $F_{X_\infty}(x)$ on an example. Queueing models are one of the most important application areas of discrete event simulation. The M/M/1 queue is known to be an analytically tractable representative of this class of simulation models.

The advantage of the M/M/1 queue is that its steady state behaviour and its transient behaviour are known, see e.g. [75-Jai91] and e.g. [79-KL85] and [92-McN91], respectively. The distribution of the number of customers in system N_i at the arrival time of the ith customer can be calculated by a numerical approach. The execution time and the memory requirements of this numerical approach allow the calculation of the distribution of the queue length of a magnitude of $i \approx 10^4$ customers on modern computers. This is usually enough to reach the steady state phase. Based on the distribution of the queue length other measures can be calculated, e.g. waiting time in queue W_i, service time S_i or the system's response time $R_i = W_i + S_i$ of the ith customer. For more details see Appendix A.3. The distributions, mean values and deciles, which are depicted in Figure 5.1, Figure 5.2 and Figure 5.3, are calculated by this numerical approach.

First we study the transient behaviour of an M/M/1 queue initialised with an empty queue and an idle server. This initialisation state is chosen very often,



(a) Evolution of deciles (dashed) and the mean (bold).



(b) $F_{R_i}(x)$ of selected customers i.

Figure 5.1: Response time of the M/M/1 queue with $\lambda = 0.95$, $\mu = 1$ assuming that the queue is initially empty.

because it is this system's state with the highest probability for $\rho < 1$. We chose $\lambda = 0.95$ and $\mu = 1$ resulting in $\rho = \frac{\lambda}{\mu} = 0.95$. Figure 5.1 shows the evolution of the distribution $F_{R_i}(x)$ of the response time towards its steady state distribution $F_{R_\infty}(x)$. In Figure 5.1(a) the mean (bold line) and the deciles (dashed lines) of $F_{R_i}(x)$ are depicted. They are strictly monotonic increasing and converging. Even though the M/M/1 queue is initialised with its state of highest probability a transient phase is present. The convergence of the mean value is similar to the convergence of the $6th$ decile. This confirms to theory,

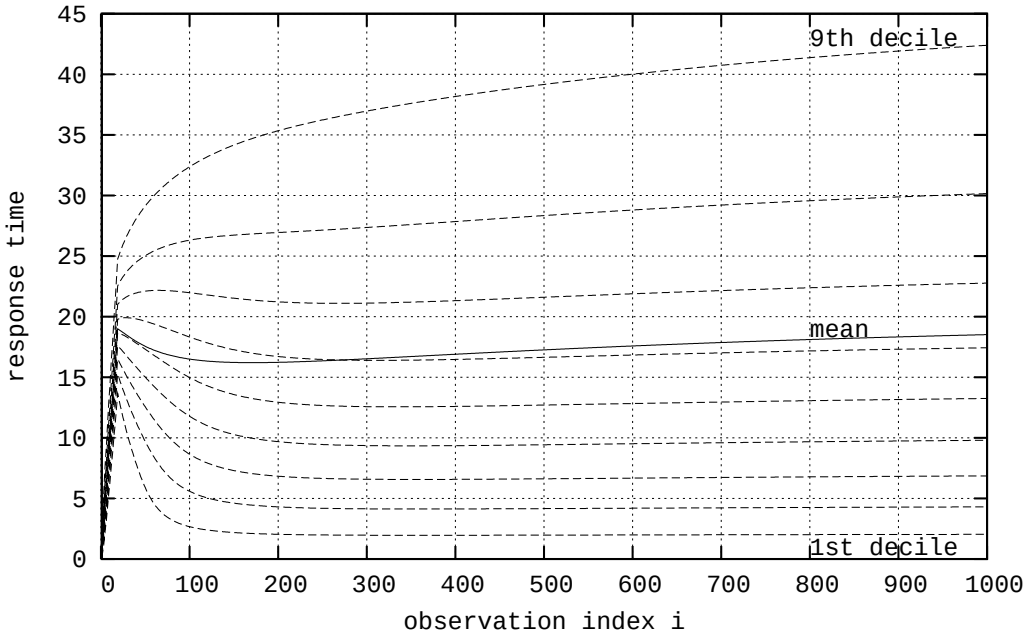
$$F_{R_\infty}(\mathrm{E}[R_\infty]) = 1 - e^{-\mathrm{E}[R_\infty]\mu(1-\rho)} \approx 0.632. \tag{5.6}$$

Smaller deciles converge faster, higher deciles converge slower in absolute sense. However, the general form of convergence looks similar for the mean and all deciles. In Figure 5.1(b) the distribution $F_{R_i}(x)$ is plotted for selected values of i (dashed graphs) and for $i = \infty$ (bold graph). From theory we know that the distribution $F_{R_i}(x)$ is a weighted sum of Erlang distributions, see Equation (A.34), and the limit response time distribution is negative exponential. Thus, in approximation we can regard $F_{R_i}(x)$ as a negative exponential distribution with increasing mean value as i is growing. Furthermore, we can approximate the convergence by a product formula which results in a scaling of the distribution of the random variable:

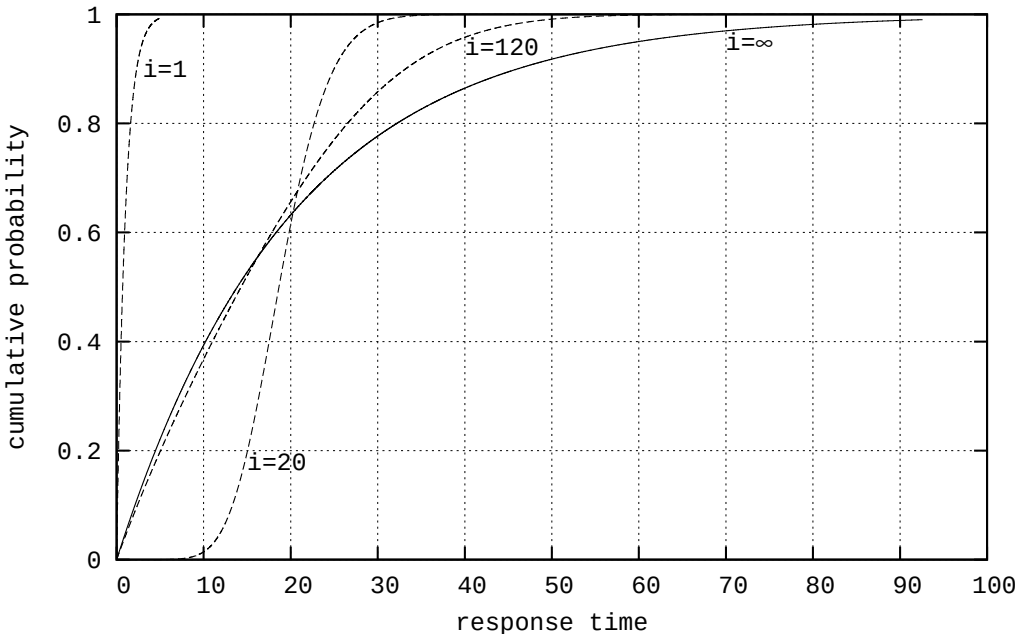
$$R_i \approx a_i \cdot R_\infty, \tag{5.7}$$

where $0 \leq a_i \leq 1$. The sequence $\{a_i\}_{i=1}^{\infty}$ converges towards the value one. Later on in this section we will see that this kind of convergence violates the preconditions of some previously known methods for the detection of l_E.

The mean number of customers in an M/M/1 queue in steady state is $\mathrm{E}[N_\infty] = \frac{\rho}{1-\rho}$. For our second example we use $\lambda = 0.95$ and $\mu = 1$ and we expect a mean number of $\mathrm{E}[N_\infty] = 19$ customers in this example. Therefore, we chose 19 initial customers. The evolution of $F_{R_i}(x)$ is depicted in Figure 5.2, including the



(a) Evolution of deciles (dashed) and the mean (bold).



(b) $F_{R_i}(x)$ of selected customers i.

Figure 5.2: Response time of the M/M/1 queue with $\lambda = 0.95$, $\mu = 1$ and 19 customers in the system at time 0.

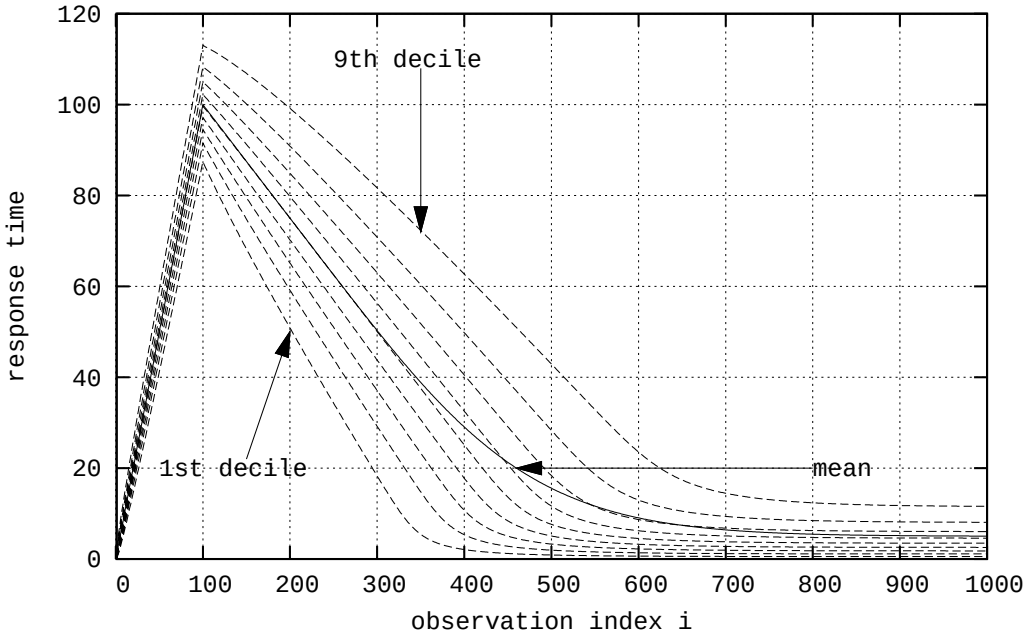
response time of the initial customers. Again, $F_{R_i}(x)$ converges towards $F_{R_\infty}(x)$. However, in Figure 5.2(a) we can see that in this case the convergence is not monotonic. This is because $F_{R_{20}}(x)$, the response time of the first non-initial customer, is completely different than $F_{R_\infty}(x)$. Both distributions are a sum of weighted Erlang distributions but in Figure 5.2(b) we can see that the density of $F_{R_{20}}(x)$ is approximately symmetric and $F_{R_\infty}(x)$ is an exponential distribution. Here, we cannot assume that the convergence can be described by a fixed class of distributions with an additional stretch or displacement. This is interesting especially for mean value analysis. It cannot be assumed that the mean value is constant right from the beginning of a simulation if the system is initialised by $\mathrm{E}[N_\infty]$ customers. This also confirms the observation of Kelton and Law that in some simple queues the optimal initial state for mean value analysis may be higher than $\mathrm{E}[N_\infty]$, see [79-KL85].

In our last example we choose an initial state, which is much higher than $\mathrm{E}[N_\infty]$. We used $\lambda = 0.8$, $\mu = 1$ and one hundred initial customers of the M/M/1 queue. Note, here we use a different setting for λ so that a much higher initial state is easier to obtain. This setting makes this example not directly comparable to the previous examples, which is not our aim. In Figure 5.3 we can see even better that the density of $F_{R_{101}}(x)$ is approximately symmetric. $F_{R_i}(x)$ is slowly converging until it is exponentially distributed at $t = \infty$. Again, no constant distribution can be assumed, but there is an obvious displacement of the mean value during the transient phase.

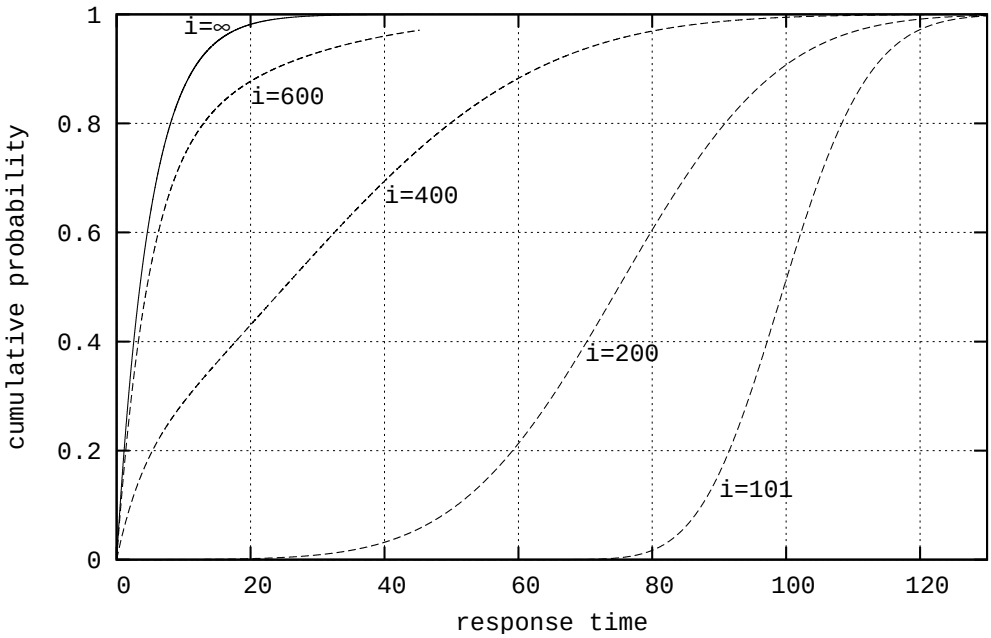
Before we step into the description of the realisation of Equation (5.1) we would like to review an assumption, which is done very often to describe transient behaviour of simulation output data. The assumption is that the transient behaviour is a displacement of a stationary process:

$$X_i = \mu_i + X'_i, \tag{5.8}$$

where the sequence $\{\mu_i\}_{i=0}^{\infty}$, with $\mu_i = \mu(1 - a_i)$, is called the transient mean



(a) Evolution of deciles (dashed) and the mean (bold).



(b) $F_{R_i}(x)$ of selected customers i.

Figure 5.3: Response time of the M/M/1 queue with $\lambda = 0.8$, $\mu = 1$ and one hundred customers in the system at time 0.

function. In [117-Sch82] is stated, that $\lim_{i\to\infty} a_i = 0$ when the simulated output process is asymptotically stationary. Furthermore, X_i' is assumed to be stationary and ϕ-mixing with a finite variance, see [117-Sch82]. In some cases X_i' is even assumed to be normally distributed, compare [118-SST83]. All these assumptions are not supported by our studies of the transient behaviour of the M/M/1 queue. It might hold for an M/M/1 queue initialised with a large number of customers, because in this situation there is an obvious displacement of the mean value. It will not hold for an M/M/1 queue which is initialised with $\mathrm{E}[N_\infty]$ customers, because here no X_i' can be found that is confirm to Equation (5.8). It will not even hold for an empty and idle initialised M/M/1 queue, a popular test model, because here the distribution of X_i' is stretched by a multiplication and not displaced by an addition. We can say that the assumed Equation (5.8) appears to be too restrictive for the M/M/1 queue. Because many queueing systems behave similar to M/M/1, this assumption seems to be not advisable for queueing systems in general. This assumption is used e.g. in [117-Sch82] to derive a test statistic based on the theory of standardised time series. We can find this assumption even in recent papers, e.g. in [3-AG04]. Here, the performance of methods based on batching observations or replicating simulations are evaluated. The results might not be valid for queueing systems in general.

For our purpose we do not make assumptions about the form of transient behaviour of an output process. Our only assumption is that the distribution function during the transient phase is different from the distribution function during the steady state phase, as described in Equation (5.1). The use of multiple independent replications with the same initial state enables us to collect an independent random sample which is distributed as $F_{X_i}(x)$. A comparison of $F_{X_i}(x)$ and $F_{X_j}(x)$ is therefore possible without mixing data of different observation indexes. For this comparison a nonparametric homogeneity test should be used that is not specialised to any family of distributions.

Based on a numerical investigation of the transient behaviour of the M/M/1 queue, which is described in [79-KL85], Kelton and Law observed that an initial number of N_0 customers, i.e. the number of customers already present in the queue at the beginning of the simulation experiment, slightly bigger than $\mathrm{E}[N_\infty]$ leads to the shortest transient phase in mean value analysis. Here, we are interested whether this is also true for analysis of quantiles or not. Figure 5.2(a) shows that the convergence of deciles is similar to the convergence of the mean. For about $i \geq 500$ the convergence of each decile seems to be monotonic. The CDF of R_∞ is given by $F_{R_\infty}(x) = 1 - e^{-x\mu(1-\rho)}$, see [75-Jai91]. The distance $\Delta(q) = |F^{-1}_{R_\infty}(q) - F^{-1}_{R_{500}}(q)|$ is therefore a valid measure of the rate of convergence of the q-quantile. We calculated $F^{-1}_{R_{500}}(q)$ for $q = \{0.1; 0.5; 0.9\}$ and $N_0 = \{16; 19; 22\}$ initial customers, with $\rho = 0.95$ and $\mu = 1$. We choose 16 and 22 initial customers to have an initialisation that is slightly smaller resp. bigger than $\mathrm{E}[N_\infty]$. To cover the full range of $0 < p < 1$ we chose two extreme quantiles and the median. The results are shown in Table 5.1, where $F^{-1}_{R_\infty}(0.1) \approx 2.107$, $F^{-1}_{R_\infty}(0.5) \approx 13.863$ and $F^{-1}_{R_\infty}(0.9) \approx 46.052$. We can see that the difference $\Delta(q)$ for an initialisation with 22 customers is smallest and, therefore, closest to the steady state results. This is not a proof, however, it indicates that Kelton and Laws observation is also true for quantiles of an M/M/1 queue. If it is true for quantiles, it must be true for the convergence of the CDF, in general. Base on these results we may assume that an initial queue length, which is slightly bigger than $\mathrm{E}[N_\infty]$, leads to the shortest transient phase for all possible measures.

$\Delta(q)$	$q = 0.1$	$q = 0.5$	$q = 0.9$
$N_0 = 16$	0.184	1.541	8.334
$N_0 = 19$	0.136	1.158	6.886
$N_0 = 22$	0.074	0.666	5.118

Table 5.1: $|F^{-1}_{R_\infty}(q) - F^{-1}_{R_{500}}(q)|$ for $q = \{0.1; 0.5; 0.9\}$ and $N_0 = \{16; 19; 22\}$

In this section we demonstrated that the output data of the simple M/M/1 queue is governed by a complex transient behaviour that is not covered by Equation (5.8). Methods, which are based on Equation (5.8), are not optimal for the M/M/1 queue and might not be for other queueing models.

5.3 Homogeneity Tests

In the discussion of the steady state phase in Section 5.1 we derived Equation (5.1). The key operation in this equation is to check the hypothesis of identically distributed random variables X_i and X_j. In the following subsections we review *nonparametric two-sample homogeneity tests* which can be used to check this hypothesis.

Let $F_{X_i}(x) = \Pr[X_i \leq x]$ be the CDF of the random variable X_i. In the *goodness-of-fit* problem the null hypothesis of

$$H_0 : F_{X_0}(x) = F_{X_1}(x) = \cdots = F_{X_{k-1}}(x) \tag{5.9}$$

is checked by a homogeneity test. A 1-sample version of a homogeneity test checks a sample of a random variable X_0 against a completely specified distribution function $F_{X_1}(x)$. Whereas in a 2-sample version the samples of two random variables X_0 and X_1 are compared with each other. Further more, a k-sample version compares k random samples with each other. In a nonparametric test there are no further assumptions about the distribution function itself. However, in general it is necessary to distinguish between the continuous and the discrete case, because this effects the test statistic.

Let $F_S(x)$ be the CDF of the test statistic. The significance level α is the probability of false rejection of the null hypothesis. Typical values are $\alpha = \{0.01, 0.05, 0.1\}$. The null hypothesis is rejected if the test statistic S is within the critical region. In a one-sided test the critical region is given by the intervals $\left[-\infty, F_S^{-1}(\alpha)\right]$ or $\left[F_S^{-1}(1-\alpha), \infty\right]$. In a two-sided test the critical region

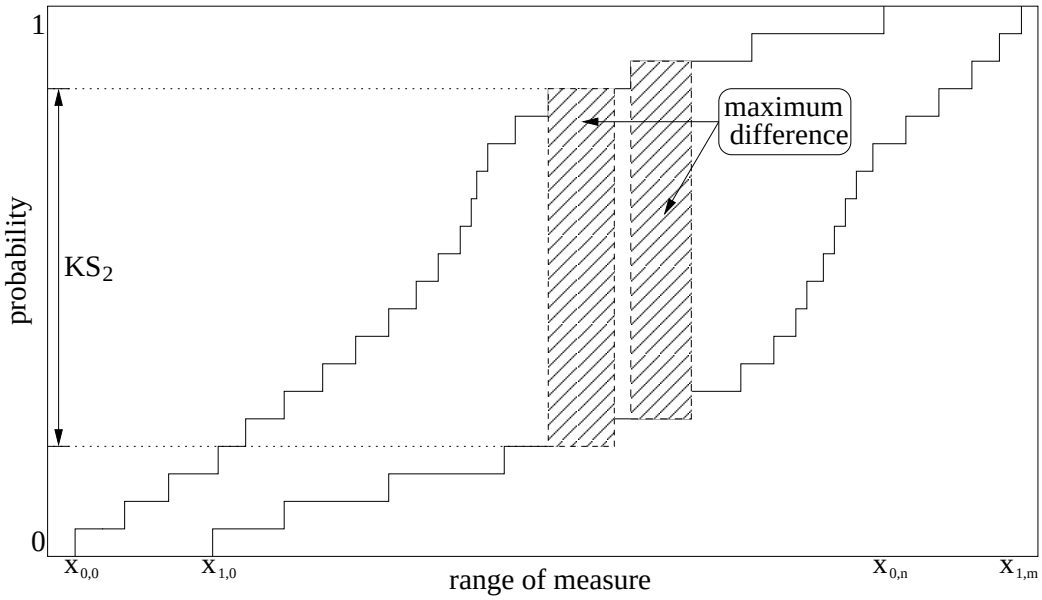


Figure 5.4: Maximum difference of two empirical distribution functions.

is given by the intervals $\left[-\infty, F_S^{-1}(\frac{\alpha}{2})\right]$ and $\left[F_S^{-1}(1-\frac{\alpha}{2}), \infty\right]$. Homogeneity test are usually one-sided with the critical value at $F_S^{-1}(1-\alpha)$.

The literature about the goodness-of-fit problem and related topics is vast, so is the number of statistical tests. Most of the tests are specialised to a given family of distributions or they are parametric. A short list of literature on non-parametric tests is [45-EJJ80], [96-NW88], [30-Dan90], [56-GC92], [122-She97] and [29-Con99]. For further discussions we choose the Kolmogorov-Smirnov test and the Anderson-Darling test because they are nonparametric. The Kolmogorov-Smirnov test is possibly the best known test and the Anderson-Darling test is maybe the most powerful one, see [126-Ste74].

5.3.1 Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov test, see [82-Kol41] and [124-Smi48], is a nonparametric homogeneity test. In the 1-sample version it is based on the statistic

$$\mathsf{KS}_1 = \sup_{-\infty<x<\infty} |\hat{F}_{X_0}(x) - F_{X_1}(x)|, \tag{5.10}$$

and in the 2-sample version it is based on the statistic

$$\mathsf{KS}_2 = \sup_{-\infty<x<\infty} |\hat{F}_{X_0}(x) - \hat{F}_{X_1}(x)|, \tag{5.11}$$

where $\hat{F}_{X_0}(x)$ and $\hat{F}_{X_1}(x)$ are the empirical CDFs of X_0 and X_1 consisting of n_0 resp. n_1 random values.

An algorithmic approach to calculate KS_2 can be based on two pointers, which are shifted within the range of the random variable. One pointer operates on the values of the random sample of X_0, the other pointer operates on X_1. This can be done by sorting the random samples of X_0 and X_1 so that two sorted sequences $\{y_{0,0}; y_{0,1}; \cdots ; y_{0,n}\}$ and $\{y_{1,0}; y_{1,1}; \cdots ; y_{1,m}\}$ are obtained, where $y_{\cdot,i} < y_{\cdot,j}$ if $i < j$. A pointer is set to the beginning of each sequence. By shifting these pointers in parallel through the interval $[\min(y_{0,0}, y_{1,0}), \max(y_{0,n}, y_{1,m})]$ the difference $\hat{F}_{X_0}(x) - \hat{F}_{X_1}(x)$ can be calculated for every value of x. Because the empirical distributions $\hat{F}_{X_0}(x)$ and $\hat{F}_{X_1}(x)$ are both step functions, only the discrete values $\{y_{0,0}; y_{0,1}; \cdots ; y_{0,n}\}$ and $\{y_{1,0}; y_{1,1}; \cdots ; y_{1,m}\}$ of x have to be regarded and x can jump from value to value in sorted order. The absolute value of the maximum of all calculated differences is the correct statistic KS_2. In Figure 5.4 the calculation of the maximum difference of the two empirical distribution functions $\hat{F}_{X_0}(x)$ and $\hat{F}_{X_1}(x)$ is demonstrated. The position of the maximum difference is within the range of the measure, whereas the test statistic KS_2 is a difference of two probabilities. Critical values for KS_1 and KS_2 are known for different α-levels and for smaller samples (< 40) they are tabulated.

5.3.2 Anderson-Darling Test

The Anderson-Darling test, see [5-AD54], is a nonparametric homogeneity test, like the Kolmogorov-Smirnov test. The original 1-sample version of the Anderson-Darling test is based on the goodness-of-fit statistic

$$\mathsf{AD}_1 = n_0 \int_{-\infty}^{\infty} \frac{(\hat{F}_{X_0}(x) - F_{X_1}(x))^2}{F_{X_1}(x)(1 - F_{X_1}(x))} \mathrm{d}F_{X_1}(x). \tag{5.12}$$

Again, $\hat{F}_{X_0}(x)$ is the empirical distribution function of X_0 consisting of random sample of size n_0. $F_{X_1}(x)$ is a completely specified distribution function. The 2-sample version of the Anderson-Darling test, see [32-Dar57] and [106-Pet76], is using the two empirical distribution functions $\hat{F}_{X_0}(x)$ and $\hat{F}_{X_1}(x)$:

$$\mathsf{AD}_2 = \frac{n_0 n_1}{n_0 + n_1} \int_{-\infty}^{\infty} \frac{(\hat{F}_{X_0}(x) - \hat{F}_{X_1}(x))^2}{H(x)(1 - H(x))} \mathrm{d}H(x) \tag{5.13}$$

with $H(x) = (n_0 \hat{F}_{X_0}(x) + n_1 \hat{F}_{X_1}(x))/(n_0 + n_1)$. In [115-SS86] and [116-SS87] the 2-sample version is extended to a k-sample version using the test statistic

$$\mathsf{AD}_\mathsf{k} = \sum_{i=0}^{k-1} n_i \int_{-\infty}^{\infty} \frac{(\hat{F}_{X_i}(x) - H'(x))^2}{H'(x)(1 - H'(x))} \mathrm{d}H'(x), \tag{5.14}$$

where n_i is the sample size of X_i and $H'(x)$ denotes the empirical distribution function of the pooled sample of all $\hat{F}_{X_i}(x)$, where $0 \leq i \leq k-1$. A computational formula for AD_k is given by

$$\mathsf{AD}_\mathsf{k} = \frac{1}{N} \sum_{i=0}^{k-1} \frac{1}{n_i} \sum_{j=1}^{N-1} \frac{(NM_{ij} - jn_i)^2}{j(N-j)}, \tag{5.15}$$

where M_{ij} is the number of observations in the sample of X_i, which are smaller or equal than Z_j. $Z_1 < Z_2 < \cdots < Z_N$ denotes the pooled and ordered sample of $H'(x)$ with $N = \sum_{i=0}^{k-1} n_i$.

In [116-SS87] is shown, that $\mathrm{E}[\mathsf{AD}_\mathsf{k}] = k-1$ holds if all $F_{X_i}(x)$ are continuous and if the null hypothesis (5.9) can be assumed. To check the null hypothesis, additionally the variance of AD_k is needed. It is given by

$$\mathrm{Var}[\mathsf{AD}_\mathsf{k}] = \frac{aN^3 + bN^2 + cN + d}{(N-1)(N-2)(N-3)}. \tag{5.16}$$

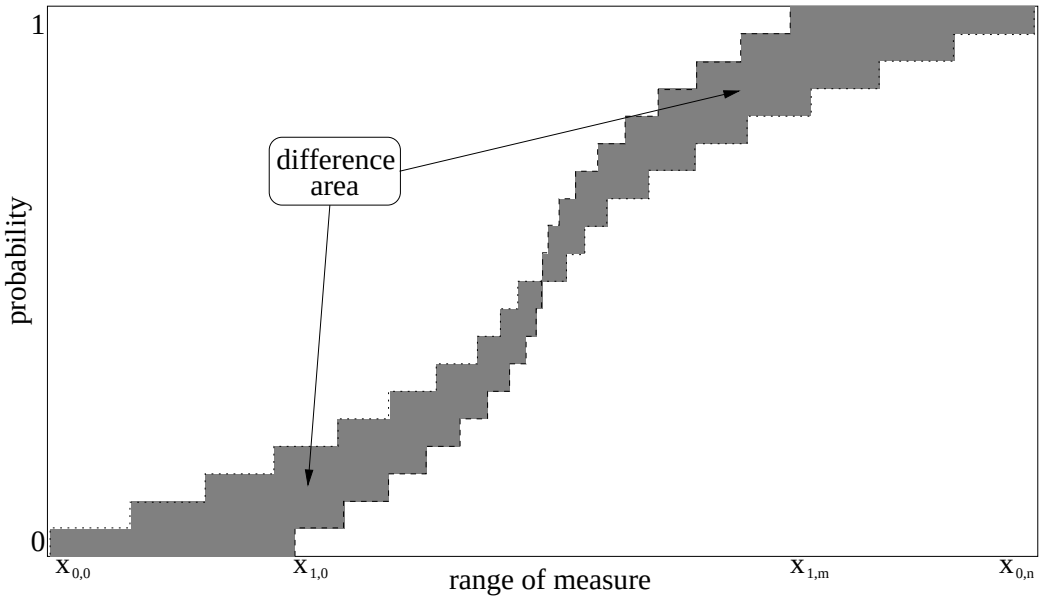


Figure 5.5: Difference area of two empirical CDFs.

For details on the calculation of a, b, c and d see [116-SS87]. $\mathsf{AD_k}$ can now be normalised by

$$T_k = \frac{\mathsf{AD_k} - \mathrm{E}\left[\mathsf{AD_k}\right]}{\mathrm{Var}\left[\mathsf{AD_k}\right]}. \tag{5.17}$$

Critical values of T_k are tabulated for $k < 12$ and various α-level. If $k \geq 12$ holds, the critical value of T_k is given by

$$t_k = b_0 + \frac{b_1}{\sqrt{k-1}} + \frac{b_2}{m}\mathrm{m}. \tag{5.18}$$

Again, values of b_0, b_1 and b_2 are tabulated for various α-level.

As it can be seen in Equation (5.14) the Anderson-Darling statistic depends on the difference of k empirical CDFs. In contrast to the Kolmogorov-Smirnov test not the maximum difference is regarded, but the integral over the whole range. This integral leads to the area between the compared empirical CDFs, compare with Figure 5.5. This area is not only influenced by the vertical difference in the range of the probability. It is also influenced by the horizontal difference in the range of the measure. Because of this, the Anderson-Darling test has a

better performance than the Kolmogorov-Smirnov test, when two distributions are compared that differ mostly at their borders, e.g. two distributions with the same expected value but different variance.

5.3.3 Accuracy

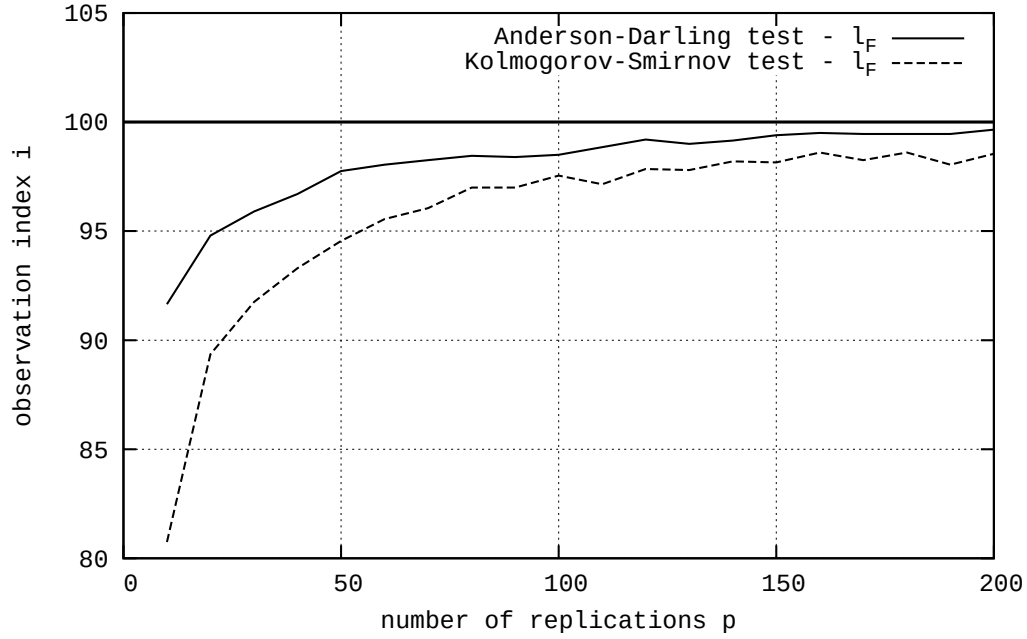
For our purpose the most interesting performance measure of the homogeneity tests is its ability to estimate l_F. Our experience with previous implementations of this method, [14-BE03] and [41-EMP05b], is that its accuracy is lower if the initial state influences mostly the tail of the density function of $F_{X_i}(x)$. For example if the mean is constant but the variance is changing over time. We believe that this problem is introduced by the KS_2 statistic, which is based on the maximum difference.

To test whether the KS_2 or the AD_k (with $k = 2$) statistic delivers better results, we applied them on two artificial output processes with a well defined truncation point l_F:

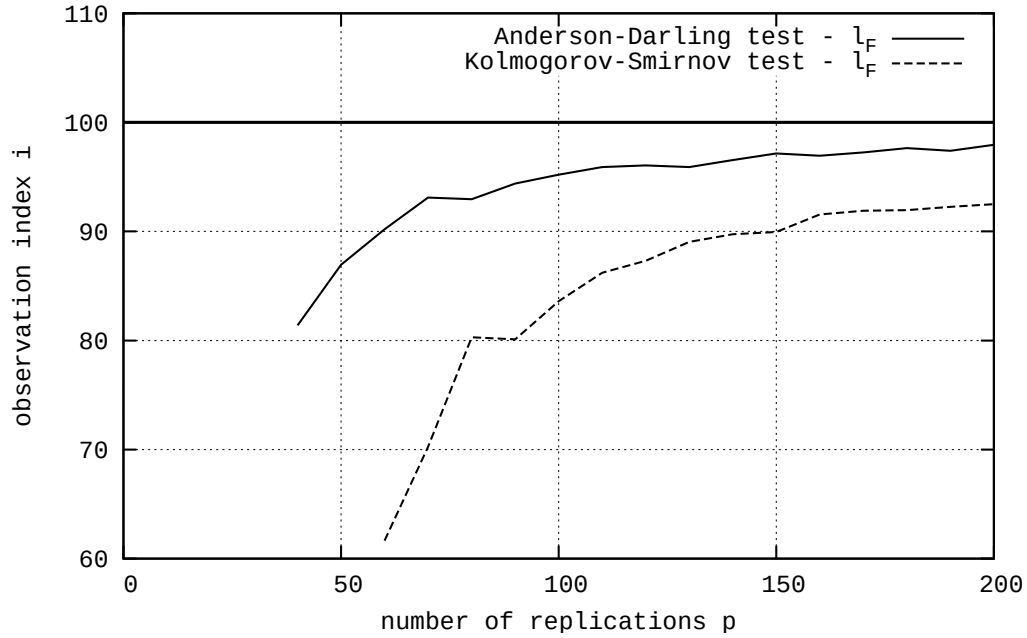
$$X_i^{(A)} = \begin{cases} \Psi_i + x - i\frac{x}{l_F} & \text{if } i < l_F, \\ \Psi_i & \text{else.} \end{cases} \tag{5.19}$$

$$X_i^{(B)} = \begin{cases} \Psi_i \cdot (x - i\frac{x-1}{l_F}) & \text{if } i < l_F, \\ \Psi_i & \text{else.} \end{cases} \tag{5.20}$$

with $x = 10$, $l_F = 100$. The randomness is introduced by the Gaussian white noise process Ψ_i with the distribution $\mathrm{N}(x; 0, 1)$. $X_i^{(A)}$ is governed by a transient mean value, whereas $X_i^{(B)}$ is governed by a transient variance. The results of the truncation point detection method, see Section 5.4, are depicted in Figure 5.6. Experiments are done for various values of $p \leq 200$. The abscissa shows p, the number of parallel replications, and the ordinate shows the estimated truncation point l_F. It is clearly evident that for both processes the estimation of l_F based on



(a) Transient Mean: $X_i^{(A)}$



(b) Transient Variance: $X_i^{(B)}$

Figure 5.6: Performance of KS_2 and AD_k: Estimated truncation point l_F (ordinate) in dependence of the sample size p (abscissa).

$\mathsf{AD_k}$ is closer to the theoretical value $l_F = 100$ when these tests are applied for our purpose. This is supported by the more general statements in [141-ZW07].

5.3.4 Time Complexity

The results of the previous section clearly suggest the use of $\mathsf{AD_k}$ instead of $\mathsf{KS_2}$. However, in practise the time complexity to calculate these statistics is another important performance measure. In the best case the calculation of $\mathsf{AD_k}$ should not require a higher computational effort. The worst case time complexities of both statistics are investigated next.

Lemma 5.3.1 *The worst case time complexity of the execution of a Kolmogorov-Smirnov 2-sample test is* $\mathrm{O}(N \log N)$ *with* $N = n_0 + n_1$.

Proof The random samples of X_0 and X_1 have to be sorted. Sorting of these two samples can be done in $\mathrm{O}(n_0 \log n_0 + n_1 \log n_1)$. Because $n_0 > 0$ and $n_1 > 0$, the inequality $n_0 \log(n_0) + n_1 \log(n_1) < N \log(N)$ holds. Therefore, the execution time of sorting can be bounded by $\mathrm{O}(N \log(N))$.

The calculation of the difference $|\hat{F}_{X_0}(x) - \hat{F}_{X_1}(x)|$ at a given value of x can be done in $\mathrm{O}(1)$, because only a constant number of basic arithmetic operations are involved. The algorithm is passing through the range of x by jumping from a $x_{i,j}$ to its successor in sorted order. Because there are $n_0 + n_1 = N$ values of x in total, the maximum difference can be calculated in $\mathrm{O}(N)$.

If the given samples are small, the critical value can be looked up in the given table in $\mathrm{O}(1)$. If the given samples are large, the critical value can be calculated by a constant number of basic arithmetic operations, which leads again to $\mathrm{O}(1)$. The comparison of the maximum difference and the critical value needs another $\mathrm{O}(1)$.

The summary of all results leads to $\mathrm{O}(N \log(N)) + \mathrm{O}(N) + \mathrm{O}(1) + \mathrm{O}(1)$. This shows, that the cardinal operation of the Kolmogorov-Smirnov 2-sample test

is the sorting of the data. Therefore, the worst case execution time is $\mathrm{O}\left(N \log(N)\right)$.

∎

Lemma 5.3.2 *The worst case time complexity of the execution of a Anderson-Darling k-sample test is* $\mathrm{O}\left(N^2 + N\log(N) + kN\right)$ *with* $N = \sum_{i=0}^{k-1} n_i$.

Proof Sorting the random samples of X_i can be done in $\mathrm{O}\left(n_i \log n_i\right)$. Consequently, sorting of all k random samples can be done in $\mathrm{O}\left(\sum_{i=0}^{k-1} n_i \log n_i\right)$. Because $\forall i(0 \leq i < k) : n_i > 0$ is valid, the overall sorting time can can be bounded by $\mathrm{O}\left(\sum_{i=0}^{k-1} n_i \log n_i\right) < \mathrm{O}\left(N \log N\right)$.

By passing in parallel through all k sorted random samples the sequence $Z_1 < Z_2 < \cdots < Z_N$ can be generated. Each value has to be accessed only once, therefore, this can be done in $\mathrm{O}\left(\sum_{i=0}^{k-1} n_i\right) = \mathrm{O}\left(N\right)$.

The ith column $\{M_{ij}\}_{j=1}^{N}$ of the M_{ij}-matrix can be calculated by passing in parallel through $\{Z_j\}_{i=j}^{N}$ and the ith sorted random sample. This is done in $\mathrm{O}\left(N + n_i\right)$. Processing all k columns leads to a run time of $\mathrm{O}\left(kN + \sum_{i=0}^{k-1} n_i\right) = \mathrm{O}\left(kN + N\right) = \mathrm{O}\left(kN\right)$.

If the M_{ij}-matrix is known, the calculation of the fraction in Equation (5.15) is done in $\mathrm{O}\left(1\right)$, because a constant number of arithmetic operations are needed. The inner sum of that equation loops over $N - 1$ values and the outer sum loops over k values. Therefore, the calculation of Equation (5.15) needs $k(N-1) \cdot \mathrm{O}\left(1\right) = \mathrm{O}\left(kN\right)$ steps.

Combining all previous results leads to $\mathrm{O}\left(N \log N\right) + \mathrm{O}\left(N\right) + \mathrm{O}\left(kN\right) + \mathrm{O}\left(kN\right) = \mathrm{O}\left(N \log N + kN\right)$, which is the overall worst case execution time to calculate the test statistic AD_k.

To normalise the test statistic AD_k its variance is needed. Here, the calculation of the parameters a, b, c and d (see Equation (5.16)) is not discussed in detail. However, the cardinal equation to calculate these parameters is

$$\sum_{i=1}^{N-2} \sum_{j=i+1}^{N-1} \frac{1}{(N-i)j}, \tag{5.21}$$

see [116-SS87]. The calculation of the fraction in Equation (5.21) is done in $\mathrm{O}(1)$. Both, the inner sum and the outer sum of that equation loop over maximum $N-2$ values. Therefore, Equation (5.21) can be calculated in $\mathrm{O}((N-2)^2) \cdot \mathrm{O}(1) = \mathrm{O}(N^2)$ steps. The calculation of T_k (see Equation (5.17)) can now be done with a constant number of basic arithmetic operations in $\mathrm{O}(1)$. Therefore, the complete normalisation can be done in $\mathrm{O}(N^2)$.

The critical value t_k can be calculated in $\mathrm{O}(1)$, no matter of the value of k. Because in every case a constant number of tabled values and basic arithmetic operations are needed. Combining all results, the overall run time of the Anderson-Darling k-sample test is given by $\mathrm{O}(N \log N + kN) + \mathrm{O}(N^2) + \mathrm{O}(1)$ $= \mathrm{O}(N^2 + kN)$. ∎

The test statistic $\mathsf{AD_k}$ depends on the difference of the empirical CDFs. In contrast to the $\mathsf{KS_2}$ statistic not only the maximum difference is used, but the integral resp. sum over the whole range of x. The higher computational complexity is caused by the calculation of $\mathrm{Var}[\mathsf{AD_k}]$, which is not depending on the data itself, but on the size of the random samples. If many Anderson-Darling tests on random samples of constant size are performed $\mathrm{Var}[\mathsf{AD_k}]$ has to be calculated only once. This is exactly the situation when performing the truncation point detection algorithm of the following sections on the output data of independent replications, because $\forall j : p_j = p$. The dominant factor in the calculation of the $\mathsf{AD_k}$ statistic itself is the sorting of the data. Therefore, the time complexity of the truncation point detection method remains the same, no matter whether the $\mathsf{KS_2}$ or the $\mathsf{AD_k}$ statistic is used. The use of $\mathsf{AD_k}$ involves an additional calculation time of $\mathrm{Var}[\mathsf{AD_k}]$ before the simulation is started. Compared to the whole run time of this method, the additional calculation time is negligible.

Corollary 5.3.3 *If $k = 2$ and $\mathrm{Var}[\mathsf{AD_k}]$ is known, than the worst case time complexity of the Anderson-Darling k-sample test and the Kolmogorov-Smirnov 2-sample test are equal.*

Proof The term $\mathrm{O}\left(N^2\right)$ is the domination factor in the worst case time complexity of the Anderson-Darling k-sample test, see Lemma 5.3.2. It is introduced due to the calculation of $\mathrm{Var}\left[\mathsf{AD}_\mathsf{k}\right]$. If $\mathrm{Var}\left[\mathsf{AD}_\mathsf{k}\right]$ is known, the dominating factor is $\mathrm{O}\left(N \log N\right)$, which is introduced due to sorting the k random samples. This leads to a reduced run time of $\mathrm{O}\left(N \log N + kN\right)$ and with $k = 2$ to $\mathrm{O}\left(N \log N + 2N\right) = \mathrm{O}\left(N \log N\right)$. This reduced run time equals the run time of the Kolmogorov-Smirnov 2-sample test, see Lemma 5.3.1. ∎

The empirical investigation of the accuracy in Section 5.3.3 shows that the estimation of l_F based on the statistic AD_k is more accurate. Thus, we suggest using AD_k instead of KS_2, see [38-Eic06].

5.4 Algorithmic Approach

In this section we show how the previously discussed 2-sample homogeneity tests can be embedded in algorithmic approaches to implement Equation (5.1). We will discuss three versions which focus on different performance measures such as precision, execution time and memory requirements. The basic idea of the algorithms is already given by Equation (5.1) and they build a new class of truncation point detection methods, which will be called the homogeneity-based truncation-point estimator in the following sections. Older truncation point detection methods only implement Equation (5.3) or Equation (5.4).

The aim is to determine the first index l_F after which all following probability distribution functions $F_{X_i}(x)$, with $l_F \leq i$, are (approximately) identical. Obviously the time horizon n of every simulation experiment is limited. Therefore it is not possible to access "all" successive probability distribution functions. Only the observed part of the steady state phase is accessible, i.e. $l_F \leq i \leq n$. It is essential that this observed part of the steady state phase is reasonably large to avoid the determination of a misleading truncation point: $l_F(r+1) = n$ with e.g.

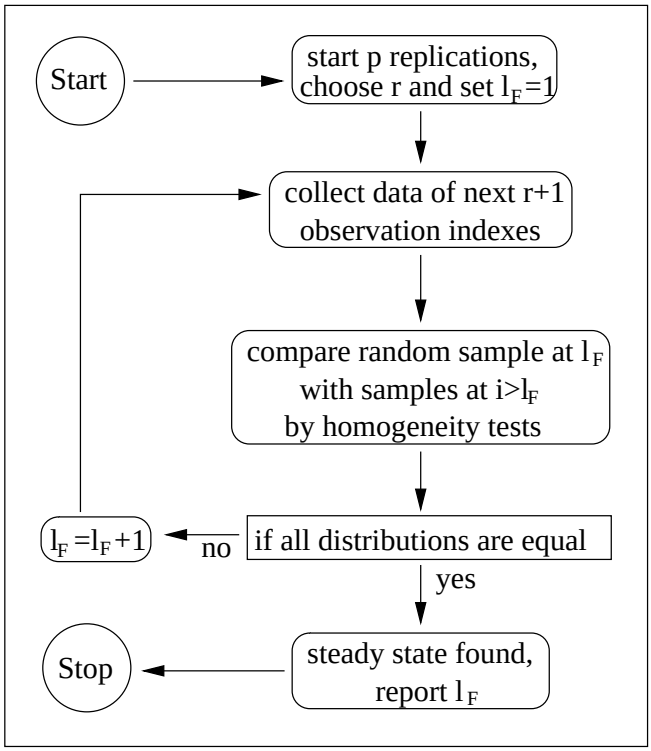


Figure 5.7: Simplified flowchart of an approach to detect l_F.

$r \geq 10$. Therefore the algorithm selects the size of the observed part of the steady state as a (constant) factor r of the size of the transient period, i.e. the size of the observed part of the steady state phase is always r-times larger than the so far selected transient phase. This is demonstrated by a flowchart in Figure 5.7.

5.4.1 Basic and Most Precise Version

This version is the most basic one, disregarding efficient execution time or low memory requirements. The algorithm is given in Listing 5.1 and was used in [13-BE02] and [14-BE03]. The pseudocode in Listing 5.1 is based on ANSI C++, but note that the operator $:=$ is an assignment and the operator $=$ is the boolean equality. The 2-sample homogeneity test is represented by the operator $\simeq$ and used to check equality in distribution of random samples of X_l and X_k. If the null hypothesis of equality is accepted by the 2-sample homogeneity test the op-

Listing 5.1: Pseudocode of the precise algorithmic approach.

```
0  int  n := 0;                  // 1 ≤ n < ∞
   int  l := 0;                  // 1 ≤ l < n
   int  r := autoSelect();       // r_min ≤ r ≤ r_max
   int  l_max := 10^5;           // l ≤ l_max

5  bool NoTestFailed := false;
   while (¬NoTestFailed){
      n := n + 1;
      observe(X_n);
      if (0 ≠ n mod (r+1)) continue;
10    l := l + 1;
      if (l > l_max) {print('Failing to detect steady state.'); break;}
      NoTestFailed := true;
      for (int k := l+1; k ≤ n; k := k+1){
         if (¬(F_{X_l}(x) ≃ F_{X_k}(x))){
15          NoTestFailed := false;
            break;
         }
      }
   }
```

erator $\simeq$ returns *true*. The variables n, l and r are defined as usual. Here, we neglect the index of the truncation point l_F to avoid and indexed variable in the pseudocode. The associated comments describe the valid range of these variables. Note, that some of the variables are initialised outside their valid range. The variable l_{max} defines the maximum range for a valid truncation point and is set to 10^5, because in our examples all estimated truncation points are smaller than $2 \cdot 10^4$ observation indexes. The method *autoSelect()* is used to find a good value for r depending on the initial part of the analysed output process and will be discussed in Section 5.5. The method *observe()* collects one observation of each of the p replications.

During each step of the algorithm the simulation horizon n is increased by one and, starting with $n = 1$, a random sample of X_n is collected. If n is a multiple of $r + 1$, l is increased by one and Equation (5.1) is checked. When the algorithm terminates $l = \{1, 2, 3, \ldots\}$ and $n = l(r + 1)$ are possible settings. Equation (5.1)

is checked by performing 2-sample homogeneity tests on the random sample of X_l compared with the random samples of X_{l+1} to $X_{l(r+1)}$. Thus, the pairs $\{X_l, X_{l+1}\}$, $\{X_l, X_{l+2}\}, \ldots, \{X_l, X_{l(r+1)}\}$ are compared by 2-sample homogeneity tests. So, a maximum number of $i \cdot r$ 2-sample homogeneity tests is performed when the loop in Line 13 of Listing 5.1 is reached for the ith time during the execution of the algorithm. If one of these tests fails l is not regarded as a valid truncation point and the algorithm is continued. For a detailed discussion on this conservative approach see Appendix A.6. Line 11 is an alternative stopping condition that prevents the algorithm to run for ever if this approach is too conservative. Otherwise l fulfils Equation (5.1) and the smallest possible truncation point is found.

Theorem 5.4.1 *The worst case execution time complexity of the algorithm in Listing 5.1 is* $\mathrm{O}\left(n^2 p \log(p)\right)$*, where* n *is the final simulation horizon and* p *is the number of parallel replications.*

Proof The loop in Line 6 of Listing 5.1 is executed n times. Therefore, Line 7 and Line 9 have a time complexity of $\mathrm{O}\left(n\right)$ in total. In Line 8 p observations are collected, this leads to a time complexity of $\mathrm{O}\left(np\right)$ in total.

Line 10 to Line 19 are only executed if the current simulation horizon is a multiple of $r+1$, this happens $\frac{n}{r+1}$ times. Thus, the total execution time complexity of Line 10 to Line 12 is $\mathrm{O}\left(\frac{n}{r+1}\right) < \mathrm{O}\left(n\right)$. During the ith execution of Line 10 to Line 19 the loop in Line 13 is performed at maximum ir times. The number of necessary 2-sample homogeneity tests, within the conditional statement in Line 14, is given by

$$r \sum_{i=1}^{\frac{n}{r+1}} i = r \frac{\frac{n}{r+1}\left(\frac{n}{r+1}+1\right)}{2} = \frac{rn^2 + r^2 n + rn}{2r^2 + 4r + 2}. \tag{5.22}$$

The only maximum of Equation (5.22) subject to $1 \leq r \leq n-1$ is at $r = \frac{n+1}{n-1}$ giving at most $\mathrm{O}\left(\frac{1}{8}(n+1)^2\right) = \mathrm{O}\left(n^2\right)$ 2-sample homogeneity tests. In Lemma 5.3.1, Lemma 5.3.2 and Corollary 5.3.3 we proved that the execution of a

2-sample homogeneity test can be done in $\mathrm{O}\left(p\log(p)\right)$. Thus, the total execution time of Line 14 is $\mathrm{O}\left(n^2 p\log(p)\right)$.

Line 15 and Line 16 are processed at maximum $\frac{n}{r+1}-1$ times because they lead to an interrupt of the loop in Line 13. Their execution time complexity is, therefore, $\mathrm{O}\left(\frac{n}{r+1}-1\right) < \mathrm{O}\left(n\right)$ in total.

The dominant factor of the time complexity of the algorithm in Listing 5.1 is introduced by Line 14 and is $\mathrm{O}\left(n^2 p\log(p)\right)$. ∎

The amount of data, which has to be processed during a simulation experiment, is usually large. Thus, as well as the worst case execution time of the algorithmic approach its storage requirement is a quite important measure.

Theorem 5.4.2 *The storage requirement of the algorithm in Listing 5.1 is* $\mathrm{O}\left(np\right)$*, where* n *is the final simulation horizon and* p *is the number of parallel replications.*

Proof In Line 8 of Listing 5.1 p real numbers are collected. We implicitly presume that these numbers are stored because they are used in Line 14 when executing the 2-sample homogeneity test. In the loop in Line 13 we access the indexes l and k, with $l+1 \leq k \leq n$. Thus, the data of all these indexes has to be stored. Only the data of the indexes with $i < l$ can be deleted. Therefore, the storage requirement is $(n-l+1)p$ which is $\mathrm{O}\left(np\right)$. ∎

In this algorithmic approach l is always shifted by only one observation index if a 2-sample homogeneity test fails. Furthermore, X_l is tested against all available X_k, with $l+1 \leq k \leq n$. This is why the algorithmic approach of Listing 5.1 is mostly precise. One disadvantage of this approach is that we approximate the final value of l strictly form below. Because all 2-sample homogeneity tests operate on a certain significance level α it is likely that the estimated l is smaller than the theoretically best choice of l. In this situation subsequent estimators, which assume identically distribution, would still be biased. Theorem 5.4.1 and Theorem 5.4.2 show that the necessary resources for the execution are high. We

recommend to use this algorithmic approach if the time to produce an observation by the simulation is significantly large.

5.4.2 Time Efficient Version

In the previous section we discussed an algorithmic approach that aims for precise estimation of l. In this section we will show that this is possible with a significantly lower execution time complexity.

Listing 5.2 shows the pseudocode of a time efficient algorithmic approach, where for convenience some special notation is used. Using the operators $+$, $-$, $/$ and $:=$ in conjunction with random variables X_k or S, see lines 4, 10, 14 and 16, means to use these operators on each component of the relevant sorted random sample separately. By S we denote a random sample $\{s_i\}_{i=1}^{p}$ of size p that is the sum of all ordered sequences which are not part of the transient period. Let $\{x_{ik}\}_{i=1}^{p}$ be the observed random sample of X_k and let $\{y_{ik}\}_{i=1}^{p}$ be the associated sorted sequence. S is then given by

$$\left\{ s_i = \sum_{k=l+1}^{n} y_{ik} \right\}_{i=1}^{p} . \tag{5.23}$$

New observations are added, see Line 10, whereas observations of the transient period are subtracted from S, see Line 14. Dividing each component of $\{s_i\}_{i=1}^{p}$ by the number of addends results in an estimate of $F_{X_\infty}(x)$:

$$\hat{F}_{X_\infty}(x) = F_{S'}(x) \quad \text{with} \quad S' \quad \text{given by} \quad \left\{ \frac{s_i}{n-l} \right\}_{i=1}^{p} . \tag{5.24}$$

The operator $\simeq$ in Line 17 and Line 21 refers to the 2-sample homogeneity test. The procedure *uniform(a,b)* delivers a uniform distributed integer random number between *a* and *b* used as index.

This algorithmic approach is similar to the one shown in Listing 5.1. However, instead of comparing X_l with all available following random samples it is compared with the averaged and standardised random sample of S', see Line 17. In

Listing 5.2: Pseudocode of the runtime efficient algorithmic approach.

```
0  int n := 0;                    // 1 ≤ n < ∞
   int l := 0;                    // 1 ≤ l < n
   int r := autoSelect();         // r_min ≤ r ≤ r_max
   int l_max := 10^5;             // l ≤ l_max
   S := 0; S' := 0; // averaged random samples
5
   bool NoTestFailed := false;
   while (¬NoTestFailed){
      n := n + 1;
      observe(X_n);
10    S := S + X_n;
      if (0 ≠ n mod (r + 1)) continue;
      l := l + 1;
      if (l > l_max) {print('Failing to detect steady state.'); break;}
      S := S − X_l;
15    S' := S/(n − l);
      NoTestFailed := true;
      if (¬(F_{X_l}(x) ≃ F_S(x))) NoTestFailed := false;
      for (int k := 1; k ≤ r; k := k + 1){
         if (¬NoTestFailed) break;
20       int u := uniform(lk + 1, l(k + 1));
         if (¬(F_{X_l}(x) ≃ F_{X_u}(x))) NoTestFailed := false;
      }
   }
```

addition X_l is compared with a random selection of r additional random samples X_u, see Line 21. These changes have a significant influence on the execution time complexity because here the maximum number of needed 2-sample homogeneity tests during each step of the algorithm is constant.

Theorem 5.4.3 *The worst case execution time complexity of the algorithm in Listing 5.2 is* $\mathrm{O}(np\log(p))$*, where* n *is the final simulation horizon and* p *is the number of parallel replications.*

Proof The run time of a single execution of Line 8 and Line 11 is $\mathrm{O}(1)$ and of Line 9 and Line 10 it is $\mathrm{O}(p)$ because p parallel replications are used. The while-loop in Line 7 is executed n times before the algorithm stops, so the run time of this part of the algorithm is $\mathrm{O}(np)$.

A single execution of Line 12 and Line 16 can be done in $\mathrm{O}(1)$. A single execution of Line 14 and Line 15 can be done in $\mathrm{O}(p)$. To execute Line 17 a

run time of $\mathrm{O}\left(p\log(p)\right)$ is needed because a 2-sample homogeneity test has to be performed, see Lemma 5.3.1, Lemma 5.3.2 and Corollary 5.3.3. Because of the condition in Line 11 this part of the algorithm is executed only $\frac{n}{r+1}$ times. Therefore, the run time of this part of the algorithm is $\mathrm{O}\left(\frac{n}{r+1}p\log(p)\right) < \mathrm{O}\left(np\log(p)\right)$.

A single execution of Line 19 and Line 20 needs only a minor run time of $\mathrm{O}\left(1\right)$. The 2-sample homogeneity test in Line 21 can be done in $\mathrm{O}\left(p\log(p)\right)$. The for-loop in Line 18 is executed at maximum r times, therefore, the run time of one complete for-loop in each step of the algorithm is $r \cdot \mathrm{O}\left(p\log(p)\right)$. All in all $\frac{n}{r+1}$ for-loops have to be performed. Therefore, the run time of this part of the algorithm is $n \cdot \frac{r}{r+1} \cdot \mathrm{O}\left(p\log(p)\right)$ which leads to $\mathrm{O}\left(np\log(p)\right)$.

Combining all results, the run time of the algorithm is $\mathrm{O}\left(np\right)+\mathrm{O}\left(np\log(p)\right)+$ $\mathrm{O}\left(np\log(p)\right) = \mathrm{O}\left(np\log(p)\right)$. ∎

Theorem 5.4.3 shows that the use of $F_{S'}\left(x\right)$ as an estimate of $F_{X_\infty}\left(x\right)$ reduces the runtime by the factor n. As we will see in the following corollary, this has no impact on the storage requirement.

Corollary 5.4.4 *The storage requirement of the algorithmic approach in Listing 5.2 is equal to the one of Listing 5.1.*

Proof To calculate S the data of the observed part of the steady state phase has to be stored, see Line 10 and Line 14 in Listing 5.2. These are the indexes $l \leq i \leq n$. Only the data of the observation indexes $i < l$ can be deleted. The amount of this data is $(n - l + 1)p$ which is $\mathrm{O}\left(np\right)$. This is equal to the storage requirement of Listing 5.1, see Theorem 5.4.2. ∎

By reducing the execution time complexity we solved one main problem of the algorithmic approach in Listing 5.1. In [41-EMP05b] is shown that this significant reduction of the time complexity has nearly no impact on the precision of the estimate l. However, the other disadvantages still remain. Because the estimate l approaches the theoretically best choice of l from below it is quite likely

that the final estimate of l is smaller than the theoretically best choice. The storage requirements are still high. We recommend this algorithmic approach if it is known that the computer can handle the amount of output data of the parallel replications.

5.4.3 Memory Efficient Version

In this section we will focus on the remaining problems of the previously introduced algorithmic approaches. The algorithmic approach that is presented in this section will have constant storage requirement. The main idea to achieve this goal is to store only a representative part of the output data. Furthermore, this algorithmic approach does not aim to estimate l as close as possible to the theoretically best truncation point. Here, the aim is to estimate a truncation point l that is not smaller than the theoretically best truncation point. This choice of l practically satisfies Equation (5.1) closer than an estimate of l that is too small, but it is maybe more wasteful.

In this algorithmic approach we split the output sequence into $r + 1$ non overlapping, equally sized and consecutive batches. During the search for a valid estimate of l we increase the size of the batches. The batch size is always increased by doubling the current batch size and so the algorithm is jumping forward geometrically. One observation index is chosen of every batch as representative. This selection is done randomly to avoid the chosen observation indexes having the same distance, because this could lead to bad performance if the output process is periodic with a constant cycle length. This approach requires the handling of some additional index pointers, which makes its source code quite long, see Listing 5.3.

batchNo is a pointer to one of the $r+1$ batches. Its valid range is $0 \leq$batchNo$\leq r$. It is used to mark the batch which is currently under observation.

batchSize is the number of observations in each batch. It is doubled after each

step of the algorithm. During the kth step of the algorithm it is 2^{k-1}.

posInBatch is a pointer to an observation index within a batch. Its valid range is $1 \leq$posInBatch$\leq$batchSize although it is initialised with zero. It is used to mark the position within the batch that is currently observed.

selectedPosInBatch is a pointer to the representative observation index within a batch. Its range is identical to the range of posInBatch and it marks the representative observation index of the current batch.

In contrast to the previous algorithms we do not implicitly assume that all X_i are stored after their collection. Here, the random sample of X_i is only stored by the conditional command in Line 16 of Listing 5.3. $b[i]$ is an array that contains observation, e.g. as floating point numbers. Its "horizontal" size is $r + 1$ entries, see Line 8. Each entry is a random sample obtained from p parallel replications, so p is its "vertical" size. The overall size of b is $p(r + 1)$.

Listing 5.3 can be separated into three parts. In the first part, Line 12 to Line 22, output samples are collected until all batches are "filled". The pointer posInBatch is increased every time a new sample is collected. The currently observed random sample is stored if posInBatch equals selectedPosInBatch. If posInBatch reaches batchSize the next batch is taken into account. This is repeated until all necessary data is collected. In the second part, Line 24 to Line 35, Equation (5.1) is checked by performing 2-sample homogeneity tests on the stored random samples $b[0]$ and $b[i]$ with $1 \leq i \leq r$. If the null hypothesis of no 2-sample homogeneity test is rejected l is set to the current simulation horizon n. Thus, possible truncation points are given by $l = n = 2^{k-1}(r + 1) = \{r + 1, 2(r + 1), 4(r + 1), \ldots\}$. This might be wasteful but it guarantees that l is deep in the steady state phase. The third part, Line 37 to Line 56, prepares b for the next step of the algorithm. Because the batch size is doubled, every two consecutive batches are combined. We have to separate between the situations

Listing 5.3: Pseudocode of the memory efficient algorithmic approach.

```
int n := 0;                          // 1 ≤ n < ∞
int l := 0;                          // 1 ≤ l < n
int r := autoSelect();               // r_min ≤ r ≤ r_max
int batchNo := 0;                    // 0 ≤ batchNo ≤ r
int batchSize := 1;                  // 1 ≤ batchSize < ∞
int posInBatch := 0;                 // 1 ≤ posInBatch ≤ batchSize
int selectedPosInBatch := 1; // see posInBatch
int l_max := 10^6;                   // l ≤ l_max
random sample b[r + 1];              // −∞ < b[i] < ∞

bool NoTestFailed := false;
while (¬NoTestFailed){
    n := n + 1;
    if (n > l_max) {print('Failing to detect steady state.'); break;}
    observe(X_n);
    posInBatch := posInBatch + 1;
    if (posInBatch = selectedPosInBatch) b[batchNo] := X_n;
    if (posInBatch = batchSize){
        batchNo := batchNo + 1;
        posInBatch := 0;
        selectedPosInBatch := uniform(1, batchSize);
    }
    if (batchNo ≤ r) continue;

    NoTestFailed := true;
    for (int i := 1; i ≤ r; i := i + 1){
        if (¬(F_{b[0]}(x) ≃ F_{b[i]}(x))){
            NoTestFailed := false;
            break;
        }
    }

    if (NoTestFailed){
        l := n;
        continue;
    }

    batchSize := batchSize · 2;
    bool ratioIsEven := (0 = r mod 2);
    int half;
    if (ratioIsEven) half = (ratio/2) − 1;
    else half = (ratio − 1)/2;

    b[0] := b[1];

    for (int i = 1; i ≤ half; i := i + 1){
      if (uniform(0,1) < 0.5) b[i] := b[i · 2];
      else b[i] := b[i · 2 + 1];
    }
    batchNo := half + 1;

    selectedPosInBatch := uniform(1, batchSize);
    if (ratioIsEven){
        if (selectedPosInBatch ≤ batchSize/2) b[batchNo] := b[batchNo · 2];
        posInBatch := batchSize/2;
    }
    else posInBatch := 0;
}
```

where the number of batches $r + 1$ is even or odd.

Theorem 5.4.5 *The worst case execution time complexity of the algorithm in Listing 5.3 is* $\mathrm{O}(np)$*, where* n *is the final simulation horizon and* p *is the number of parallel replications.*

Proof The loop in Line 11 of Listing 5.3 is executed n times. Line 12, Line 15 and Line 17 to Line 22 cause only a minor runtime of $\mathrm{O}(n)$ in total. In Line 14 p, observations are collected which leads to a runtime of $\mathrm{O}(np)$ in total. In Line 16 p, observations are stored. This is done once for all $r + 1$ batches for all k steps of the algorithm. Thus, the runtime of this line is $\mathrm{O}(krp)$.

Because of the condition in Line 22 consecutive lines are executed only once for every step of the algorithm. Note, that $n = 2^{k-1}(r + 1)$. Therefore, $k = \log\left(\frac{n}{r+1}\right) + 1$ steps are performed to reach the final simulation horizon n. The most interesting lines beyond Line 22 are Line 26, Line 46 and Line 47. All other lines cause only a minor runtime.

In Line 26, r 2-sample homogeneity tests are executed in every step. A 2-sample homogeneity test can be done in $\mathrm{O}(p\log(p))$, see Lemma 5.3.1, Lemma 5.3.2 and Corollary 5.3.3. Thus, the total runtime of all 2-sample homogeneity tests is $\mathrm{O}(krp\log(p))$.

In Line 46 and Line 47 less than r random samples of size p are moved in the memory in every step. This causes a runtime of $\mathrm{O}(krp)$.

Combining all results we receive a runtime of $\mathrm{O}(np + krp\log(p))$. We can replace $\mathrm{O}(k) = \mathrm{O}\left(\log(\frac{n}{r+1})\right) < \mathrm{O}(\log(n))$. Because r is a positive constant and $r << n$ usually holds, we can conclude $\mathrm{O}(np + rp\log(p)\log(n)) = \mathrm{O}(np)$. ∎

This proof shows that, surprisingly, the collection of the output data in Line 14 is the most time consuming part of the algorithm in Listing 5.3. Other parts of the algorithm must only be taken into account if a truncation point is found already

after a few steps, i.e. if k is small and, therefore, $r << n$ does not hold. However, in this situation the runtime of the algorithm might not be of interest at all.

This algorithmic approach uses the least time for any given step because the number of 2-sample homogeneity tests is constant, given by r. It takes the least number of steps to terminate because checkpoints are spaced geometrically with $2^{k-1}(r+1)$. The geometrical spacing of checkpoints leads to a truncation point that is deeper in steady state. In the worst case, this truncation point is twice as large as the one determined by Listing 5.1 and Listing 5.2. The computational burden of the additional simulation needs to be accounted for, however, a factor 2 is negligible for the runtime in sense of the $\mathrm{O}(.)$-notation.

Theorem 5.4.6 *The storage requirement of the algorithm in Listing 5.3 is* $\mathrm{O}(rp)$*, where* r *is the selected ratio between the transient phase and the observed part of the steady state phase and* p *is the number of parallel replications.*

Proof All the variables n, l, r, batchNo, batchSize, posInBatch and selectedPosInBatch need $\mathrm{O}(1)$ memory. The array b can store $r+1$ random samples of size p. When a random sample of X_i is collected in Line 14 it is only stored, if this is explicitly stated by Line 16. Because r and p are constant parameters during the execution of Listing 5.3 the memory requirements are constantly $\mathrm{O}(rp)$. ∎

Theorem 5.4.5 and Theorem 5.4.6 show that the algorithmic approach of Listing 5.3 is most efficient. It can be used if a large truncation point is expected because its storage requirements are constant and its runtime is superior over other algorithmic approaches. The estimated l is not as small as possible because this algorithm is shifting l with geometrically growing step size. Furthermore, the final estimate of l is set to the simulation horizon n. This makes this approach more wasteful than the algorithms of Listing 5.1 and Listing 5.2. However, it fulfils Equation (5.1) because the estimate l is deeper in steady state than necessary. The other algorithms estimate l as small as possible but tend to underestimate it. If the influence of the initial state is only present up to a well defined observation index,

the estimate l of algorithm of Listing 5.3 might even be able to fulfil Equation (5.1) with strict equality instead of approximate equality.

5.5 Parameterisation

The transient behaviour of a simulation output process $\{X_i\}_{i=0}^{\infty}$ can have many different forms. Therefore, it is almost impossible to find a parameterisation of an estimator that works well for all kinds of output processes. If an estimator is limited to a certain class of models, e.g. queueing models or time series, this might be possible. The purpose of automated and sequential simulation is to deliver estimates with a small error for a previously unknown process. This implies that the analyst cannot provide any information about $\{X_i\}_{i=0}^{\infty}$ and has no further knowledge of a valid parameterisation of an estimator. If there is no standard parameterisation available and the analyst cannot give a valid parameterisation the parameters of the estimator must be set automatically.

5.5.1 Parameter r

The only critical parameter of the algorithms presented in Listing 5.1, Listing 5.2 and Listing 5.3 is the ratio r between the length of the transient phase and the length of the observed part of the steady state phase. If r is chosen too small the algorithms may overlook transient behaviour. This is especially critical if the convergence of the output process towards its steady state behaviour is very slow. In general r should not be smaller than a certain threshold r_{min}. If r is chosen too big the algorithms will require an unnecessarily long runtime. In general r does not need to be greater than a value r_{max}. As with r itself, optimum values for r_{min} and r_{max} depend on the output process.

r should be chosen large enough to overcome the serial correlation of the output process. The serial correlation coefficients c_k of a single server queue are analysed in [31-Dar68] and the calculation of c_k for the M/M/1 queue is shown.

Listing 5.4: Pseudocode of the automated selection of parameter r.

```
0  int r := 0;            // r = 0 or r_min ≤ r ≤ r_max
   int r_min := 10;  // 1 ≤ r_min ≤ r_max
   int r_max := 1000;  // r_min ≤ r_max < ∞
   int n := 0;            // 1 ≤ n < ∞

5  while (n − 1 ≤ r_max){
       n := n + 1;
       observe(X_n);
       if (n − 1 < r_min) continue;
       if (¬(F_X1(x) ≃ F_Xn(x))){
10         r := n − 1;
           break;
       }
   }
   if (r = 0) print('No transient behaviour detected.');
```

Comparing $\sum_{k=0}^{k_{max}} c_k$ for $k_{max} = 10^i$ with $k_{max} = \infty$ should give a rough idea of how to choose r_{min} and r_{max}. In Table 5.2 the results of an M/M/1 queue at low and high traffic intensity ρ is shown. The value of k_{max} is set to different orders of magnitude. For $\rho = 0.5$ most of the correlation is within a lag of $k_{max} = 10^1$, therefore r should not be smaller than $r_{min} = 10$. And for $\rho = 0.9$ most of the correlation is within a lag of $k_{max} = 10^3$, thus we set $r_{max} = 10^3$. All experiments in Section 5.6 are done with this setting of r_{min} and r_{max} leading to good results. Because it is possible to find a parameterisation of r_{min} and r_{max} that is valid for many models, these parameters are not critical.

To find a valid choice of r itself, we compare $F_{X_1}(x)$ with $F_{X_n}(x)$ for increasing values of n with respect to the limits $r_{min} \leq r \leq r_{max}$. A valid ratio $r = n-1$ is found if $F_{X_1}(x) \neq F_{X_n}(x)$ holds, because the transient behaviour will not be overlooked. If no $r \leq r_{max}$ can be found, we assume that the process is stable

k_{max}	10^0	10^1	10^2	10^3	10^4	∞
$\rho = 0.5$	1.778	4.598	5.333	5.333	5.333	5.333
$\rho = 0.9$	1.991	10.521	71.858	178.454	181.818	181.818

Table 5.2: Sum of the first k_{max} correlation coefficients of the M/M/1 queue.

right from the beginning. Pseudocode of this approach is given in Listing 5.4.

This algorithm is used to initialise the parameter r for the algorithms in Listing 5.1, Listing 5.2 and Listing 5.3. Furthermore, we assume that all observed random samples of X_n, see Line 8, are stored and reused by the truncation point detection algorithm. Especially for the algorithm in Listing 5.3 this data can be used to initialise the array b. The runtime and the storage requirements of the algorithm in Listing 5.4 are obviously lower than the runtime and the storage requirements of the truncation point detection algorithms.

5.5.2 Parameters of the Homogeneity Test

The previously discussed homogeneity tests operate at a certain significance level α. We do not consider α as a critical parameter because e.g. $\alpha = 0.05$ is a valid setting for all kinds of output processes. For the multiple comparisons during one step of the algorithms we chose the most conservative approach by demanding that no 2-sample homogeneity test should reject the null hypothesis. Compare with the discussion in Appendix A.6.

Another parameter of the homogeneity tests is the size p of the random samples. Here, this is given by the number of parallel replications. Figure 5.6 in Section 5.3.3 shows the dependence of the homogeneity tests on p. We can see that the graphs flatten out at about $p = 100$, which is therefore a good setting. We also see that $p < 30$ or even $p < 50$ is not a good choice, the estimated truncation point is too far away from its optimum. Because of these observations it is possible to choose a setting of p, e.g. $p = 100$, that is valid for many kinds of output processes and p is not a critical parameter.

5.6 Validation and Comparison

In this section experiments on the previously introduced truncation point detection methods are done. The experiments are based on a large variety of models to cover many different kinds of output processes. The simulation results are compared with the results of other well known methods, which are discussed next. For a survey on this topic see [99-Paw90].

Crossing of the Mean is not a statistical test in the classical sense, i.e. no null hypothesis is tested. We follow [99-Paw90] and regard it as a rule of thumb. Let

$$\bar{X}_k = \frac{1}{k}\sum_{j=1}^{k} X_j \tag{5.25}$$

be the mean of the first k observations X_1 to X_k. Define

$$c_{j,k} = \begin{cases} 1 & \text{if}(X_j > \bar{X}_k \text{ and } X_{j+1} < \bar{X}_k) \text{ or } (X_j < \bar{X}_k \text{ and } X_{j+1} > \bar{X}_k), \\ 0 & \text{else,} \end{cases} \tag{5.26}$$

with $1 \leq j \leq k-1$. The value

$$c_k = \sum_{j=1}^{k-1} c_{j,k} \tag{5.27}$$

shows how many times the sequence $\{X_1, \ldots, X_k\}$ is crossing the mean $\bar{X}_k$. In [49-Fis73] is pointed out that this rule "may prove useful in practise for assessing the dilution over time of bias due to initial conditions". A test can be applied by checking the condition $c_k \geq c$, where c is a critical value defined by the analyst. If the condition is not fulfilled for k, another observation is added to the sequence of data and the test is performed for $k+1$. As pointed out before, this test does not check a null hypothesis, therefore, no significance level is given. This makes the selection of the critical value c difficult. An optimal setting for c depends on the output process itself. In [54-GAM78] the performance of this test is evaluated for examples and the critical value is recommended to be $c = 25$. We used this setting

for all our experiments. The crossing of the mean rule focuses on a constant mean, so we can say that it is a realisation of Equation (5.3).

Combined stationarity tests for proving if a time series of a fixed size has a transient behaviour are given in [64-GSS94] and [22-CDL$^+$92] and the earlier publications [117-Sch82] and [118-SST83]. The earlier published tests are based on the assumption that the transient behaviour of the output process can be described by Equation (5.8). As we already pointed out in Section 5.2, this assumption is quite strict because only a displacement of the distribution (addition of a value) is covered. In the more recent publications Equation (5.8) is replaced by just assuming a transient mean function defined by

$$\mu_i = \mathrm{E}[X_i] = \mu(1 - a_i), \tag{5.28}$$

where the a_i's are constants. This is more general because no stationarity of X_i' is assumed anymore. However, in [64-GSS94] it is additionally assumed that the variance of the sample mean during the initial transient ($i < l_E$) is greater than the variance of the sample mean during steady state ($i \geq l_E$). This does not hold in general, and especially for queueing systems this assumption about the variance is critical. Compare this with the discussion of the convergence of the response time of an empty and idle initialised M/M/1 queue in Section 5.2. In this situation $\mathrm{Var}[R_i] < \mathrm{Var}[R_{i+\delta}]$ holds for $\delta > 0$, see Figure 5.1(a). Furthermore, Equation (5.28) does not cover the situation of a constant mean but a transient process variance, i.e. $\mathrm{E}[X_i] = \mathrm{E}[X_{i+\delta}]$ but $\mathrm{Var}[X_i] \neq \mathrm{Var}[X_{i+\delta}]$. This might not cause a problem when the only measure of interest is $\mathrm{E}[X_\infty]$. It does cause a problem when e.g. quantiles of the distribution function $F_{X_\infty}(x)$ are of interest. Examples of this kind of output processes are given later in this section and they are used for our experimental studies.

In contrast to crossing of the mean the combined stationarity tests are based on the variance of the sample mean. The sample of output data with fixed size n is split into two windows $X_1, \ldots, X_{n'}$ and $X_{n'+1}, \ldots, X_n$. The variance σ_1^2

of the sample mean of the first window ($i < n'$) is tested against the variance σ_2^2 of the second window ($i \geq n'$). Different estimators for σ_1^2 and σ_2^2 are discussed in [64-GSS94], such as the batch means estimator, the area estimator, the maximum estimator and combinations of these. Because of the transient mean function $\sigma_1^2 > \sigma_2^2$ is assumed. Therefore, σ_1^2/σ_2^2 is tested against the F-distribution at $1 - \alpha$ and parameters κ_1 and κ_2. α is the significance level and κ_1 and κ_2 are the degrees of freedom in the estimation of σ_1^2 and σ_2^2. As already discussed, we cannot assume this is valid for all possible output processes. For the area and the maximum estimator, see [22-CDL$^+$92], which are part of the combined stationarity tests, the asymptotic variance needs to be estimated. We used the spectral variance estimator, which is described in [72-HW81] and in [93-MEP04], with a constant window size of 200 observations. This window was placed at the end of our original sequence. The combined method is proposed for the purpose of mean value analysis. However, we will show that this method can provide valid estimates of l_V, see Equation (5.4), for selected examples and if the test statistic σ_1^2/σ_2^2 is adjusted to σ_2^2/σ_1^2 or even to a two sided test.

Our focus is on sequential output analysis. Therefore, we embed this statistical test in an algorithmic approach, as it was done with the homogeneity test in Section 5.4. Firstly, we introduce a distance between the two windows. This may shift the second window deeper into the steady state phase and the difference between σ_1^2 and σ_2^2 might be more obvious. Secondly, if we fail to find a valid truncation point we always discard the oldest observation and add a new observation X_{n+1}. In this way the analysed sequence has a constant size. However, the shift of only one observation leads to a large computational effort. Our focus here is to receive a precise truncation point, the complexity of the runtime is a secondary issue.

Because we are interested in automated analysis, we are looking for a set of parameters that is valid for a wide range of output processes. Furthermore, we assume that there is no additional information about the analysed model than the

observations itself. We should keep in mind, that universality is one of the main advantages of simulation. If more information is given about the analysed model, it might make more sense to use a different analysis technique than simulation. In [22-CDL$^+$92] some settings for the parameters are evaluated on examples. Based on those investigations we choose 16 batches in the first and second window and a distance of 10^3 observations between both windows. The correlation within the output sequences of our various test models is very different, therefore, we were not able to choose an optimal batch size m that is valid for the whole set of our test models. Depending on the model we used $m = \{3, 10, 50, 100, 200\}$, if not stated explicitly we set $m = 10$. All our experiments are performed at $\alpha = 0.05$.

The Homogeneity-Based Truncation-Point Estimator is described in Section 5.4. To obtain a precise estimate of the truncation point Listing 5.1 is used in Sections 5.6.1, 5.6.2, 5.6.3 and 5.6.4. The ratio between the current transient observations and the current observed part of the steady state is chosen automatically by the algorithm itself, see Section 5.5. The significance level of the homogeneity test, i.e. the Anderson-Darling test, is set to $\alpha = 0.05$ for all experiments.

In the experiments in Section 5.6.7 we have a different aim. Here, our aim is not to detect a precise truncation point but to choose a truncation point large enough, so that the data is identically distributed, but without being too wasteful. We will see that in this case it is not necessary to use exact versions of the truncation point detection methods. Here, we apply Listing 5.3. Again, for this method the ratio between the so far detected transient observations and the so far observed part of the steady state is chosen automatically by the algorithm itself and we used $\alpha = 0.05$. Results of this method are tested against the results of a combination of the crossing of the mean rule and Schruben's test, see [118-SST83], which is implemented in Akaroa2, see [47-EPM99], and we use the standard settings of this software tool. These are: 25 crossings of the mean; $\gamma = 0.5$; $\gamma_v = 2$; varianceLength$= 100$; $\alpha = 0.05$ and safetyFactor$= 1$. For further details on this

parameters the reader is referred to the original publications.

All Experiments are based on at least 10^4 replications. This means that for the crossing of the mean rule and the combined stationarity test all experiments are repeated at least 10^4 times. We will see that this leads to smooth graphs of empirical distribution functions. Here, the homogeneity-based truncation-point estimator is always based on 100 replications. Therefore, experiments with this method are repeated at least 100 times to make these results comparable to the results of the other methods. The results are used to calculate the mean, standard deviation and more derived measures, as well as the empirical CDF and a histogram of the density function. We use the random number generator described in [87-LSCK02]. Distribution functions are constructed with the help of the software library *dcdflib*, which implements methods of [1-AS65]. If the evolution of a process is depicted it is always based on 101 replications and the quantiles $q = \{0.066, 0.184, 0.332, 0.5, 0.668, 0.816, 0.934\}$, see [40-EMP05a]. Details to the depiction of quantiles evolving over time can be read in Chapter 4.

If the bias of subsequent estimators is mentioned in the following sections, we refer to the remaining initialisation bias after deleting a sequence of data in the beginning. If the data beyond the truncation point is not identically distributed this causes bias in the estimators, which assume identically distributed data.

5.6.1 Basic Models

The first experiments are done by analysing the output data of a model with no initial transient phase, and of two unstable models. Any truncation point detection method should return $l = 1$ ($l_F = l_E = l_V$) for the first model and it should be able to recognise that the unstable models are not converging towards a steady state measure, i.e. $l_F = l_E = \infty$ for the second and $l_F = l_V = \infty$ for the third model.

Figure 5.8 is a plot of a selection of quantiles of the output process evolving

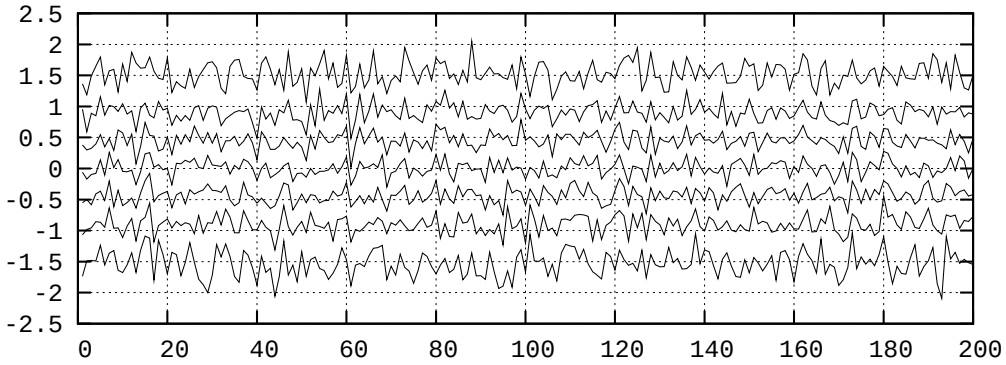


Figure 5.8: Quantiles of the Gaussian white noise process evolving over time.

over time. The first model is the *Gaussian white noise process*

$$X_i = \Psi_i, \tag{5.29}$$

where the CDF of every Ψ_i is given by the standard normal distribution $F_{\Psi_i}(x) = \mathrm{N}(x; 0, 1)$. F_{Ψ_i} is constant over time, therefore, the output process is stable from the beginning. The theoretically best truncation point is at $l = 1$. Despite of small changes at low frequency, no general trend can be seen. Table 5.3 shows the result of the crossing of the mean rule, the combined stationarity test and the homogeneity-based truncation-point estimator. *Mean* is the average of all simulation results. *Standard error* of the mean is the standard deviation divided by the square root of the number simulation results. *Standard deviation* is the square root of the variance of all simulation results. Due to the selection of the seed of our random number generator the results of each experiment can be regarded as

	crossing	combined	homogeneity
mean	50.33 ± 0.07	1.78 ± 0.04	1.37 ± 0.08
std. dev.	6.99	4.46	0.84
min.	31	1	1
max.	74	105	7

Table 5.3: Simulation results of the Gaussian white noise process.

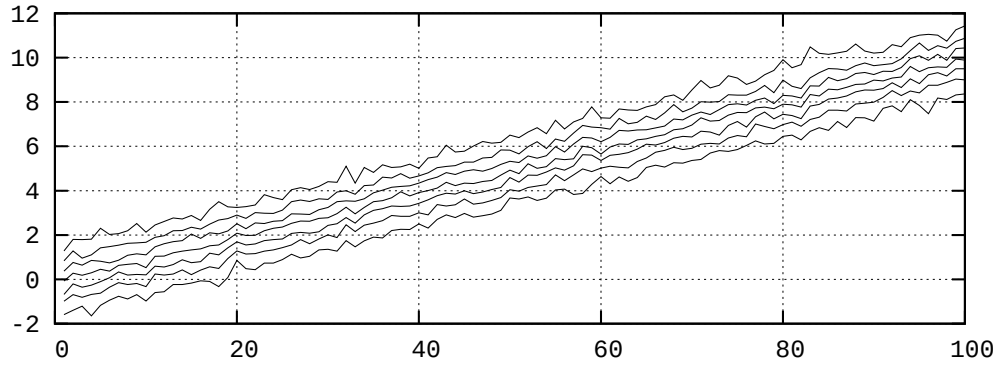


Figure 5.9: Quantiles evolving over time of the unstable model with increasing mean.

independent and identically distributed. Thus, the calculation of the variance can be done as usual. *Minimum* and *maximum* are the lowest and the highest result. These measures are based on the distribution function of the truncation point estimator and show how precise a truncation point can be estimated by a method. The results of the homogeneity-based truncation-point estimator are best, in the sense that the mean is closest to $l = 1$ and the standard deviation is smallest. The results of the combined stationarity test are similar, even though the mean and the standard deviation are a little bit bigger. The crossing of the mean rule could not detect the truncation point precisely, and even the minimum of all results is too high.

The second output process is unstable, because its mean value is constantly increasing with simulation time.

$$X_i = (c \cdot i) + \Psi_i, \tag{5.30}$$

where $c = 0.1$ is a constant slope. Figure 5.9 shows that in this case the quantiles of the output process are displaced. The displacement is increasing over time. Longer simulation experiments would show even higher values. In Table 5.4 the simulation results of the unstable model with increasing mean are listed. The

percentage *unstable* shows for how many runs the detection method recognised that the model is unstable. All methods have been set to assume that the model is unstable if the truncation point is greater than a certain threshold. Such a threshold is needed as otherwise they would try to find a truncation point without being able to fulfil the stopping condition and run forever. Usually we set this threshold at 10^5 observations, however, to avoid long execution times of the simulation runs we reduced this parameter for the experiments with unstable models. Here, all methods assume that the model is unstable if the truncation point is greater than 10^3. The crossing of the mean rule and the homogeneity-based truncation-point estimator are able to detect that the output values are not converging to a steady state value. We clearly see that the combined stationarity test is not able to detect this. The two estimated values for the sequence variance σ_1^2 and σ_2^2, which are calculated in the first and the second window of the combined stationarity test, are identical, because the variance estimator is independent of the displacement. Therefore, the combined stationarity test is not sensitive for a displacement if it appears in both windows.

The third output process is unstable, because its variance is constantly increasing with increasing simulation time.

$$X_i = (c \cdot i) \cdot \Psi_i, \tag{5.31}$$

where $c = 0.1$ is a constant slope. Figure 5.10 shows that in this case the quantiles of the output process are stretched. The stretch is increasing over time. Longer

	crossing	combined	homogeneity
unstable	96%	0%	100%
mean	-	1.114 ± 0.005	-
std. dev.	-	0.46	-
min.	-	1	-
max.	-	7	-

Table 5.4: Simulation results of the unstable model with increasing mean.

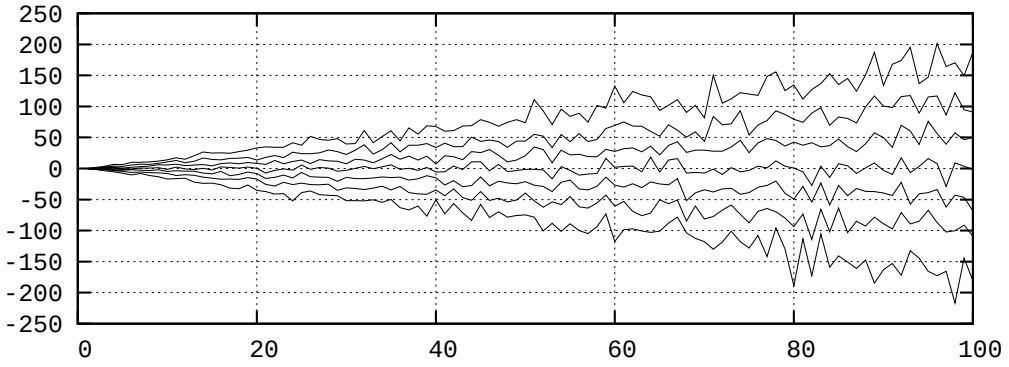


Figure 5.10: Quantiles evolving over time of the unstable model with increasing variance.

simulation experiments would show even higher values. Usually the combined stationarity test assumes that the sequence variance is smaller in the steady state phase than during the transient phase, i.e. $\sigma_1^2 > \sigma_2^2$. Here, we know that $\sigma_1^2 < \sigma_2^2$ and we adjusted the combined stationarity test by testing σ_2^2/σ_1^2 against the F-distribution. Table 5.5 shows, that the adjusted combined stationarity test and the homogeneity-based truncation-point estimator were able to detect the instability of the model. Not surprisingly, the crossing of the mean rule failed. This method assumes that an output process is stable if the mean is constant over time. The assumption does not hold for this model.

	crossing	combined	homogeneity
unstable	0%	99%	100%
mean	52.81 ± 0.08	-	-
std. dev.	7.51	-	-
min.	33	-	-
max.	91	-	-

Table 5.5: Simulation results of the unstable model with increasing variance.

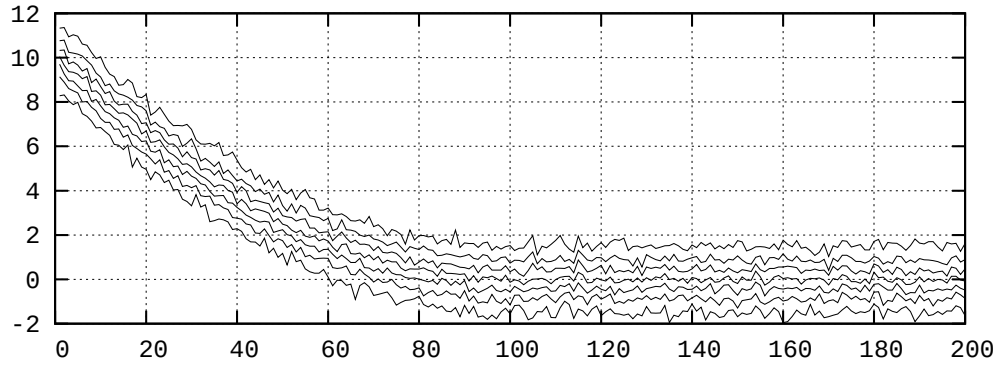


Figure 5.11: Evolution of quantiles of an output process with parabola displacement in the beginning.

5.6.2 Transient Mean Value

In this section experiments are done with transient output processes. All output processes converge towards a steady state distribution. In addition, during the transient phase $\mathrm{E}[X_i] \neq \mathrm{E}[X_j]$ holds if $i \neq j$. This condition is important for all truncation point detection methods, which are based on the convergence of $\mathrm{E}[X_i]$.

The first process shows an initial quadratic displacement:

$$X_i = \begin{cases} \Psi_i + \frac{k}{l^2}(l-i)^2 & \text{if } i < l, \\ \Psi_i & \text{else,} \end{cases} \tag{5.32}$$

where k is the offset. It has got a well defined truncation point l, where $l_F = l_E = l$ and $l_V = 1$. $\mathrm{E}[X_i]$ is governed by a parabola for $i < l$. This can be seen in

	crossing	combined	homogeneity
mean	198.4 ± 0.2	61.71 ± 0.07	90.2 ± 0.3
std. dev.	20.3	7.12	3.39
min.	122	44	81
max.	252	140	104

Table 5.6: Simulation results of the output process with parabola displacement in the beginning.

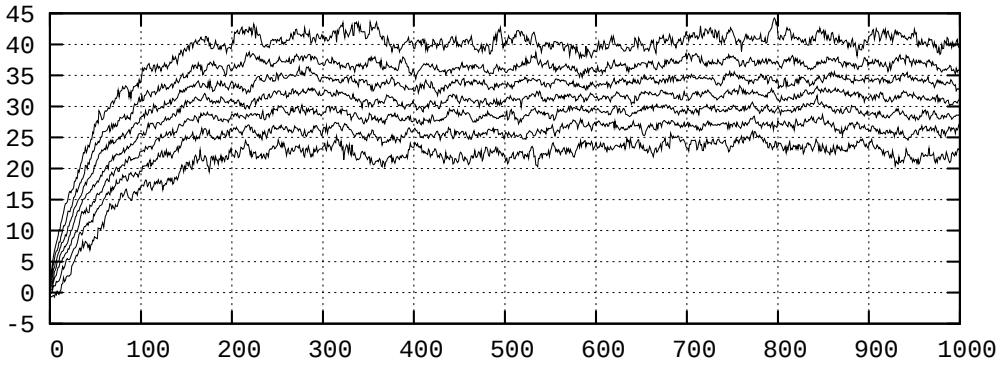


Figure 5.12: Evolution of quantiles of a geometrical ARMA$(5, 5)$ process.

Figure 5.11. In this example and for all simulation experiments with this model we chose $k = 10$ and $l = 100$. We performed the crossing of the mean rule, the combined stationarity test and the homogeneity-based truncation-point estimator on this output process. The simulation results are listed in Table 5.6. The results of the homogeneity-based truncation-point estimator are closest to the theoretically best result. Furthermore, the standard deviation of these results is smallest, which indicates a stable estimate. However, the majority of the results is smaller than the theoretically best truncation point. If the truncation point is too small, the data will still be biased. The coverage of estimators in a subsequent analysis will be reduced by this. The impact of this bias is even stronger when using the combined method. All simulation results of this method are smaller than the theoretically best truncation point. All simulation results of the crossing of the mean rule are beyond the theoretically best truncation point. From point of view of an accurate truncation point detection, the results of this method are the worst. However, the results are already deep in the steady state phase and thus a bias on subsequent estimators is avoided in this example.

The next output process is given by a geometrical ARMA$(5, 5)$ process, see

Appendix A.2.

$$\Upsilon_i^{(5)} = 1 + \Psi_i + \sum_{j=1}^{5} \frac{1}{2^j}(\Upsilon_{i-j}^{(5)} + \Psi_{i-j}). \tag{5.33}$$

We chose $\Upsilon_i^{(5)} = 0$ for $i \leq 0$. In Figure 5.12 the evolution of the quantiles of this process can be seen. $\mathrm{E}\left[\Upsilon_i^{(5)}\right]$ is converging towards the value $\mathrm{E}\left[\Upsilon_\infty^{(5)}\right] = 32$. After about 200 observation indexes the process seems to be stable. For this process we adjusted the test statistic of the combined stationarity test. We tested σ_1^2/σ_2^2 and σ_2^2/σ_1^2 against the F-distribution, both at $1 - \frac{\alpha}{2}$. For this 2-sided test only one critical value of the F-distribution needs to be calculated. Alternatively, σ_1^2/σ_2^2 could be tested against the F-distribution at $1 - \frac{\alpha}{2}$ and $\frac{\alpha}{2}$. Previous experiment series with less replications showed that this two sided test is more powerful than each 1-sided test on its own. The simulation results for this geometrical ARMA$(5, 5)$ process are listed in Table 5.7. Comparing this simulation results with Figure 5.12 we see that the results of the crossing of the mean rule are too large. The results of the combined stationarity test are too small. Only the results of the homogeneity-based truncation-point estimator are located as expected, around observation index 200. Because all estimated truncation points of the crossing of the mean rule are deep in the steady state phase, this method is wasteful, but again we can conclude, that no bias would affect subsequent estimators.

The next output process is a damped vibration.

$$X_i = \Psi_i + (ke^{i\frac{\ln(0.05)}{l}}) \cdot \cos(\omega i), \tag{5.34}$$

where k is the amplitude and $T = \frac{2\pi}{\omega}$ is the cycle length. k is damped by an

	crossing	combined	homogeneity
mean	458 ± 2	65 ± 1	216 ± 4
std. dev.	143	59	39.8
min.	135	1	152
max.	983	594	355

Table 5.7: Simulation results of a geometrical ARMA$(5, 5)$ process.

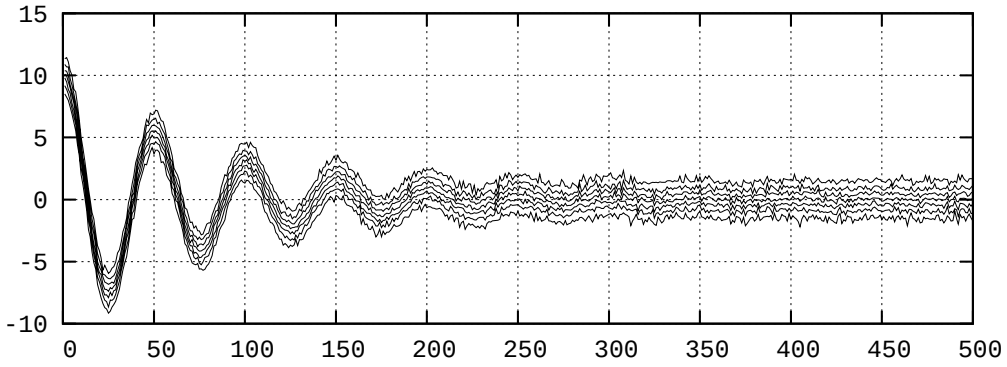


Figure 5.13: Evolution of quantiles of the damped vibration.

exponential function. Here, $l_F = l_E = l$ and $l_V = 1$. At $i = l$, the exponential function is 0.05, therefore, a truncation point that is greater than l can be regarded as a suitable truncation point for reasons of comparison. In our experiments we used $k = 10$, $T = 50$ and $l = 250$. The evolution of the quantiles of this process is depicted in Figure 5.13. The crossing of the mean rule is a simple heuristic. The results in Table 5.8 show, that it does not work very well if the convergence is not monotone. In this case all results of this method are too small. Bias would be introduced in subsequent methods, which assume identically distributed data. The results of the combined stationarity test are similar. However the maximum result of this method seems to be deep enough in the steady state phase. The homogeneity-based truncation-point estimator is the only method that returns useful results in this example, even though the standard deviation of the es-

	crossing	combined	homogeneity
mean	175.3 ± 0.2	183.9 ± 0.3	353 ± 22
std. dev.	14.2	26.1	215
min.	133	124	284
max.	226	341	2275

Table 5.8: Simulation results of the damped vibration.

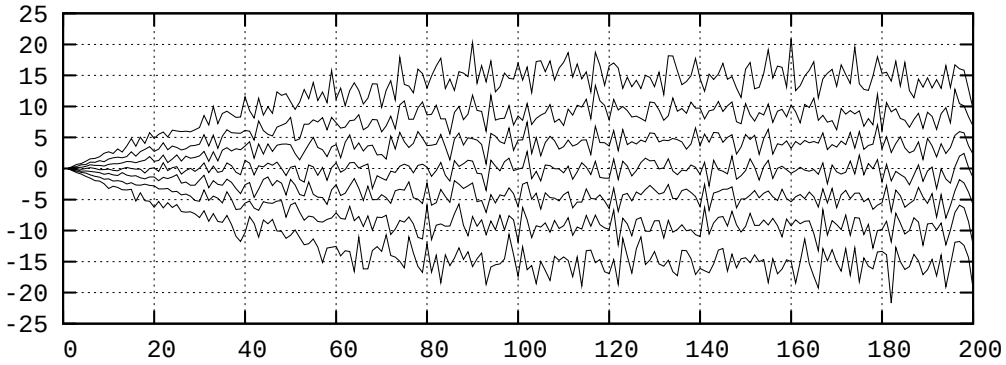


Figure 5.14: Evolution of quantiles of a process with parabola stretch in the beginning.

timated truncation points is high. This is due to the long tail of the distribution of the estimated truncation points toward infinity, compare minimum and maximum value in Table 5.8.

5.6.3 Constant Mean Value

In contrast to the previous section we now demonstrate experiment series with output processes that have got a constant mean, i.e. $\mathrm{E}[X_i] = \mathrm{E}[X_j]$ holds for any i and j. However, all the output processes we use here do not have a constant distribution, because $F_{X_i}(x) \neq F_{X_j}(x)$ holds for $i \neq j$ during the transient phase. Detection methods, which are specialised to determine a truncation point l_E, might fail determining a valid truncation point l_F, as discussed in Section 5.1.

The first process is governed by a quadratic stretch of the distribution function

	crossing	combined	homogeneity
mean	52.45 ± 0.08	32.2 ± 0.1	58 ± 1
std. deviation	7.53	14.1	10.5
min	33	1	42
max	88	118	125

Table 5.9: Simulation results of the process with parabola stretch in the beginning.

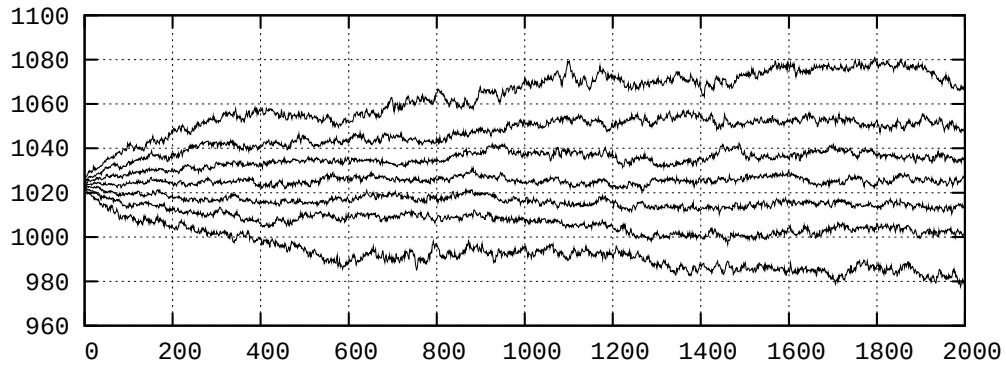


Figure 5.15: Evolution of quantiles of a geometrical ARMA$(10, 10)$ process.

during the transient phase.

$$X_i = \begin{cases} (2i\frac{k}{l} - i^2\frac{k}{l^2})\Psi_i & \text{if } i < l, \\ k\Psi_i & \text{else,} \end{cases} \tag{5.35}$$

where k is the final stretch during the steady state phase and l, where $l_F = l_V = l$ and $l_E = 1$, is the theoretically best truncation point. Here we chose $k = 10$ and $l = 100$. Figure 5.14 shows the evolution of a selection of quantiles of this process. In Table 5.9 the results of the simulation experiments with this process are listed. Again we know that the variance of the process is higher during steady state and so we tested σ_2^2/σ_1^2 against the F-distribution in the combined stationarity test. A batch size of $m = 3$ is adequate for this model. For this model the estimates of the homogeneity-based truncation-point estimator are not as close to the theoretically best truncation point as the results for the parabola displacement. This

	crossing	combined	homogeneity
mean	450 ± 3	156 ± 3	620 ± 22
std. deviation	293	284	216
min	52	1	211
max	3392	2026	1333

Table 5.10: Simulation results of a geometrical ARMA$(10, 10)$ process.

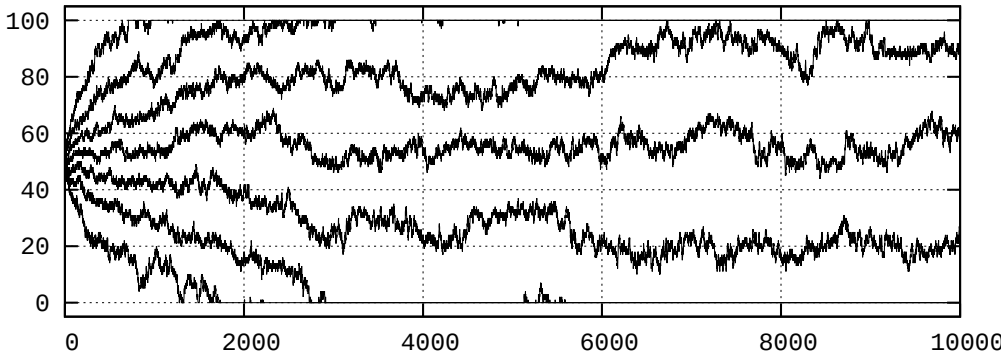


Figure 5.16: Evolution of quantiles of the bounded random walk.

indicates that it is more difficult for the homogeneity test to distinguish between distributions with a different variance than between distributions with different mean. However, the minimum and maximum values show that the range of the estimates is placed around $l = 100$. The range of the estimates of the combined stationarity test also include $l = 100$, but the minimum truncation point of this method is the first observation. Furthermore, the mean truncation point of this method is too small. The mean truncation point of the crossing of the mean rule is higher than that of the combined stationarity test, but the minimum and maximum estimate show that the results of this method are also too small. From point of view of mean value analysis the results of the combined stationarity test and the crossing of the mean rule are acceptable, because $\mathrm{E}[X_i]$ is constant for all i. However, these two methods are not able to detect the truncation point from which on $F_{X_i}(x)$ is constant. The homogeneity-based truncation-point estimator is the only method that can be recommended, if other measures than $\mathrm{E}[X_\infty]$ are of interest.

The next output process is again a geometrical ARMA$(10, 10)$ process, similar as defined in Equation (5.33). Here we chose $\Upsilon_i^{(10)} = \mathrm{E}\left[\Upsilon_\infty^{(10)}\right] = 1024$ for $i \leq 0$. Because of this, $\mathrm{E}\left[\Upsilon_i^{(10)}\right] = 1024$ for $1 \leq i < \infty$ is constant right from the beginning, other measures of the distribution are not. The evolution of

the quantiles of this process is shown in Figure 5.15. In Table 5.10 the results of the simulation experiments with this process are listed. As with the previous geometrical ARMA$(5, 5)$ process we have to adjust the test statistic of the combined stationarity test. We tested σ_2^2/σ_1^2 against the F-distribution and used a batch size of $m = 50$ because of the highly correlated data. The mean truncation point of the homogeneity-based truncation-point estimator is deepest in the steady state phase, followed by the mean truncation point of the crossing of the mean rule and then the one of the combined stationarity test. The standard deviation of the results of all methods is high, which leads to high maximum values for the crossing of the mean rule and the combined stationarity test.

The next process is based on a random walk X'_i, which is defined by

$$X'_i = \begin{cases} X'_{i-1} + 1, & \text{with probability } 0.5, \\ X'_{i-1} - 1, & \text{with probability } 0.5, \end{cases}$$

with the initial state $X'_0 = 50$. The process X'_i can take any value between $-\infty$ and $+\infty$. The final process X_i is bounded, so that its range is the interval $[0, 100]$:

$$X_i = \begin{cases} 0, & \text{if } X'_i < 0, \\ X'_i, & \text{if } 0 \leq X'_i \leq 100, \\ 100, & \text{if } X'_i > 100. \end{cases}$$

Because X_i is bounded a marginal distribution for $i = \infty$ exists. The peculiarity of this process is that the expected value $\mathrm{E}[X_i] = 50$ is constant over i, whereas all quantiles other than the median are not constant and converge to the thresholds

	crossing	combined	homogeneity
mean	384 ± 5	355 ± 9	8494 ± 259
std. deviation	441	884	2586
min	40	1	3210
max	7647	7761	17624

Table 5.11: Simulation results of the bounded random walk.

0 and 100. This can be seen in Figure 5.16. $F_{X_i}(x)$ is very steep around $x = 50$ for small i. After a long simulation time the shape of $F_{X_i}(x)$ is completely different. For large i it is very flat around $x = 50$. However, the expected value $\mathrm{E}[X_i]$ is constant for all i. Analysis of mean values only would show a constant behaviour, even though this process is transient and the CDF is slowly converging to its marginal distribution. Again, for the combined stationarity test we chose σ_2^2/σ_1^2 and tested it against the F-distribution because the variance is increasing over time. We set a batch size of $m = 200$ because of the extremely correlated data. Comparing the simulation results listed in Table 5.11 with Figure 5.16 we clearly see, that the estimated truncation points of the crossing of the mean rule and the combined stationarity test are too small. After deletion the remaining data is still not identically distributed. Only the results of the homogeneity-based truncation-point estimator seem to be reasonably deep in the steady state phase.

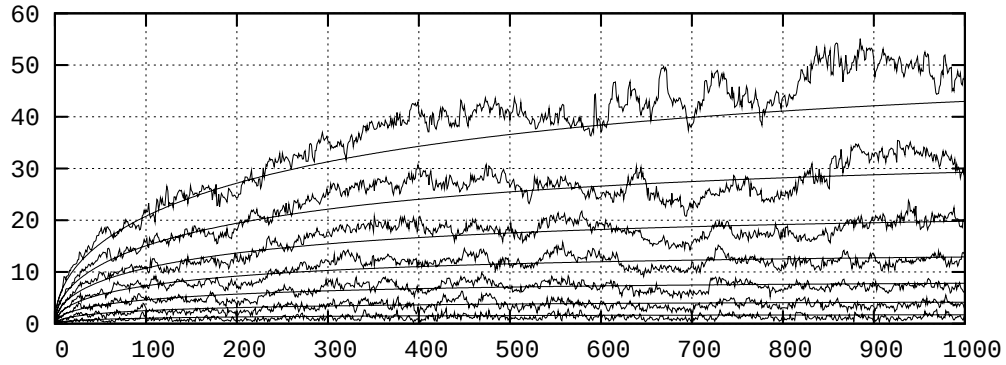


Figure 5.17: Evolution of quantiles of the M/M/1 queue without initial customers.

5.6.4 Queueing Models

Queueing models are a very important application of stochastic discrete event simulation. Here we use the output data of a selection of standard examples for our simulation experiment series.

The next analysed output stream is the time in system of a job leaving an M/M/1 queue with arrival rate $\lambda = 0.95$ and service rate $\mu = 1$. In consequence the traffic intensity of this server is $\rho = 0.95$ and it is stable. No initial customers are waiting in the queue. In Figure 5.17 the evolution of a selection of quantiles is depicted. The graphs with high frequency changes are based on 101 simulation runs, whereas the flat lines are the theoretical course of the quantiles calculated by the method in Appendix A.3. Because $\rho = 0.95$ we expect highly correlated data and chose a batch size of $m = 100$ for the combined stationar-

	crossing	combined	homogeneity
mean	366 ± 3	359 ± 7	456 ± 22
std. deviation	259	694	215
min	51	1	188
max	2280	5235	1328

Table 5.12: Simulation results of the M/M/1 queue without initial customers.

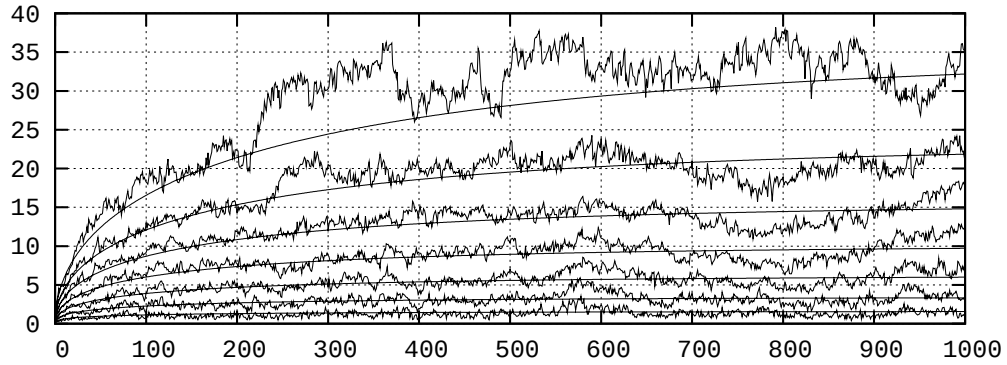


Figure 5.18: Evolution of quantiles of the M/E_2/1 queue.

ity test. The values listed in Table 5.12 are the simulation results of the M/M/1 queue. The mean truncation point of the homogeneity-based truncation-point estimator is deepest in the steady state phase. The range of the results of the combined method and the crossing of the mean rule are larger than the range of the results of the homogeneity-based truncation-point estimator. However, the results of all of these methods seem to be too small to eliminate the initialisation bias completely. Figure 5.17 shows that the higher quantiles have not converged to their steady state value at e.g. the 456th observation index.

The next used queueing model is an M/E_2/1 queue. We chose the interarrival rate $\lambda = 1$ and the mean service rate of the Erlang distribution $\mu = \frac{1}{0.95}$ with the shape (number of stages) 2. This queueing model is stable because the traffic intensity is $\rho = 0.95$. The evolution of the quantiles is depicted in Figure 5.18 and

	crossing	combined	homogeneity
mean	345 ± 2	351 ± 7	406 ± 25
std. deviation	236	682	249
min	55	1	115
max	2559	4565	1646

Table 5.13: Simulation results of the M/E_2/1 queue.

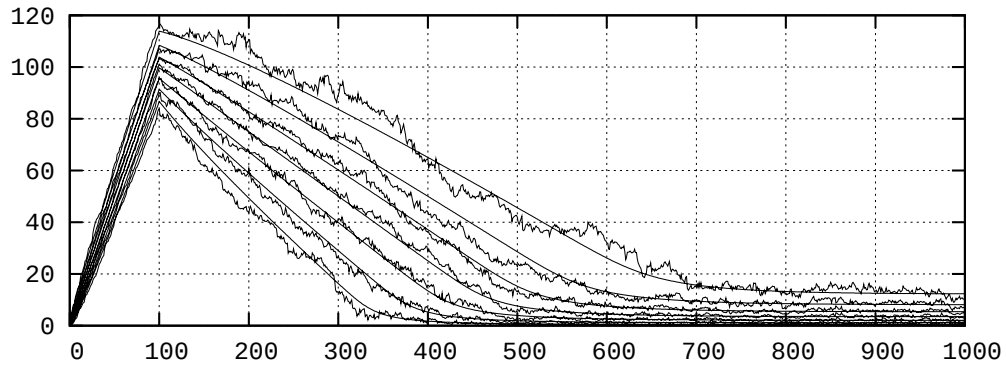


Figure 5.19: Evolution of quantiles of the M/M/1 queue with initial customers.

results are shown in Table 5.13. Again, the theoretical course and the estimation of the quantiles based on 101 replication can be seen. We expect highly correlated data and chose a batch size of $m = 100$ for the combined stationarity test. The results of the M/E_2/1 queue are very similar to the results of the M/M/1 queue. The mean truncation point of the homogeneity-based truncation-point estimator is deepest in the steady state phase and within the smallest range.

The next model is again an M/M/1 queue. In contrast to the previous experiments, here, we chose $\lambda = 0.8$ and $\mu = 1$ resulting in $\rho = 0.8$. Furthermore, the initial queue length is one hundred customers. The evolution of the quantiles is depicted in Figure 5.19. Again, the theoretical course and the estimation of the quantiles based on 101 replication can be seen. Because of the initial customers the plot shows a non-monotonic convergence of the response time towards

	crossing	combined	homogeneity
mean	1940 ± 8	401 ± 2	725 ± 48
std. deviation	789	180	480
min	259	32	573
max	5362	1480	6209

Table 5.14: Simulation results of the M/M/1 queue with initial customers.

its steady state distribution. The mean truncation point of the crossing of the mean rule is deepest in the steady state phase. However, comparing the results of Table 5.14 with Figure 5.19 we can see that the crossing of the mean rule is too wasteful. The mean truncation point of the combined method appears to be too small because most quantiles have not converged to their steady state value. The mean truncation point of the homogeneity-based truncation-point estimator is located in an area where quantiles seem to have converged.

5.6.5 Distribution of the Truncation Points

The results presented so far are derived measures of the distribution of the estimated truncation points. We get a deeper insight if we have a look at the CDF and the probability density function (PDF) of the estimates. Therefore, we depict empirical CDF and histograms of the previous simulation results of selected models in Figures 5.20 to 5.24, which can be found at the end of this section. These simulation experiments are based on at least 10^4 replications. The graphs of the homogeneity-based truncation-point estimator are not as smooth as the graphs of the combined stationarity test because in the last case one estimate is based on 100 replications which leads to a lower number of final estimates.

In Figure 5.20 the empirical CDF and an histogram of the estimated truncation points of the combined stationarity test and the homogeneity-based truncation-point estimator for the process with the quadratic displacement are shown, see Equation (5.32) and compare with Table 5.6. We can see that both histograms show an almost symmetric distribution. The right tail of the distribution is short and bounded because this process has got a well defined truncation point. The left tail is short and greater than zero because the transient behaviour is obvious for both test methods. Again we can see that the results of the homogeneity-based truncation-point estimator are closer to the theoretical optimum than the results of the combined stationarity test. However, the symmetrical form of both

distributions shows that the estimates are stable.

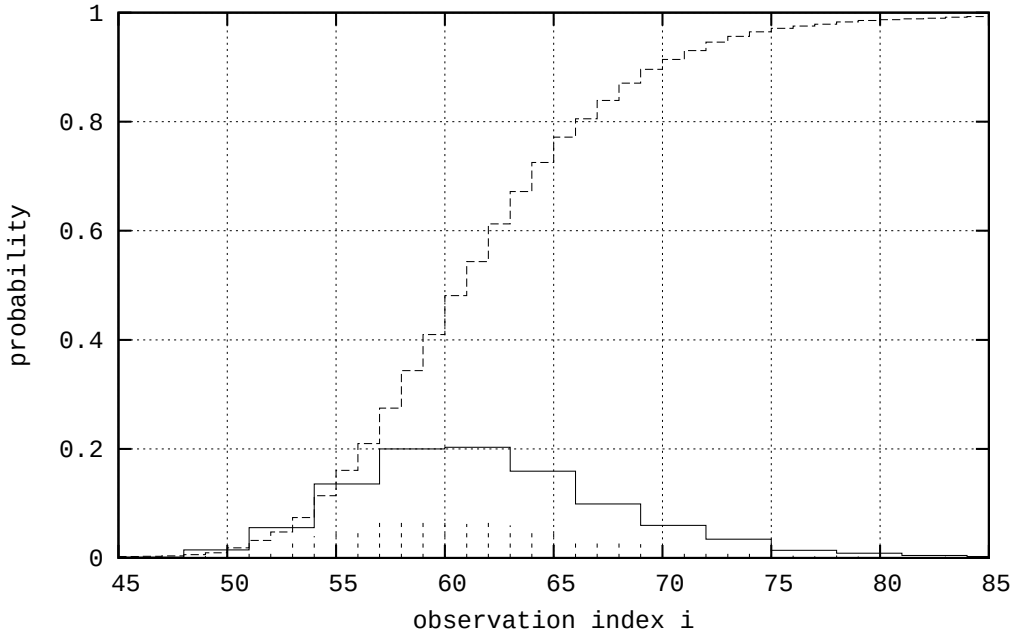
In Figure 5.21 the distribution of the results for the geometrical ARMA$(5, 5)$ process and $\Upsilon_i^{(5)} = 0$ for $i \leq 0$ are depicted, see Equation (5.33) and compare with Table 5.7. Both distributions are not symmetric. They have a long right tail because this process has no well defined truncation point and the initial state theoretically influences the process for all i. The left tail of the distribution of the combined stationarity test even includes zero. This shows that the output stream of some replications could meet the conditions of the combined stationarity test from the beginning. The left tail of the distribution of the homogeneity-based truncation-point estimator is bounded at around 150 observations and, therefore, this test guarantees the deletion of transient observations.

The distributions for the process with the damped vibration, see Equation (5.32) and compare with Table 5.6, is quite interesting. Because of its non-monotonic transient behaviour the distribution of the estimated truncation points is multi-modal. The maxima, resp. minima, of the amplitudes of the damped vibration are directly visible in the distribution of the homogeneity-based truncation-point estimator. Whenever the process, i.e. the test sample, is close to a maximum or minimum it is unlikely that the homogeneity-based truncation-point estimator detects a truncation point. The maxima and minima of our analysed output process are located at $\frac{1}{4}kT$ with the cycle length $T = 50$ and integer value k. Compare this locations with the observation indexes 275, 300, 325 and 350 in Figure 5.22(b). In the distribution of the results of the combined stationarity test these maxima and minima locations are only indirectly visible. This method is more likely to reject the hypothesis of identically distribution if the number of maxima and minima within the first and second window is unequal, compare Figure 5.22(a).

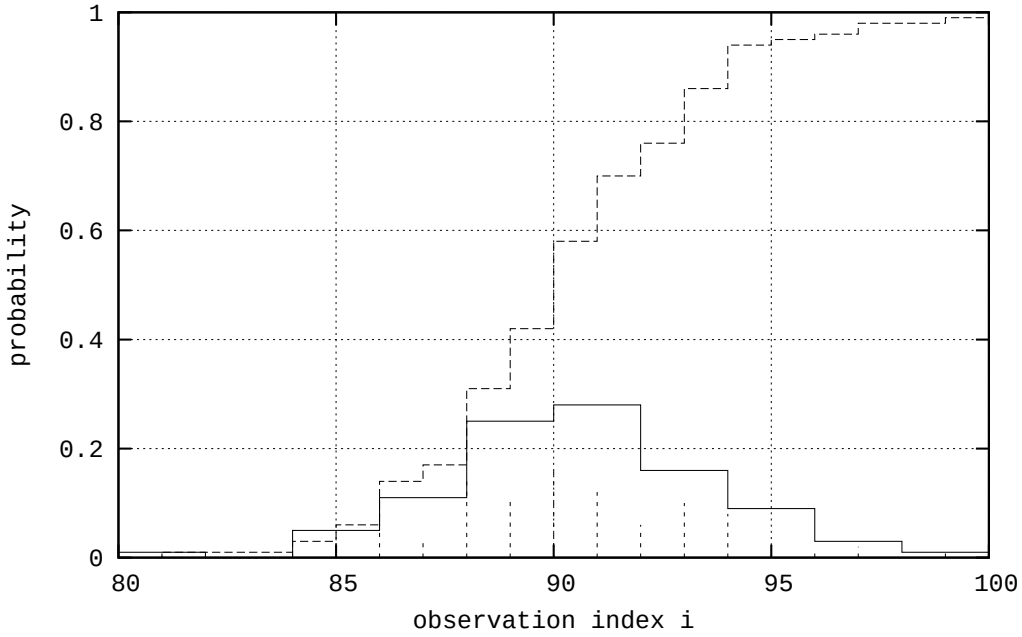
In Figure 5.23 the distribution of the estimated truncation points of the M/M/1 queue with $\rho = 0.95$ and no initial customer is depicted, compare with Table 5.12. Surprisingly, the distribution of the homogeneity-based truncation-point estimator

and the combined stationarity test look completely different. The distribution of the homogeneity-based truncation-point estimator is comparable to the distribution of the ARMA process in Figure 5.21(b). The form of the tails seems to be similar, even though the mode is not a clearly visible. In contrast, the distribution of the results of the combined stationarity test shows a high mode at observation index one and a very long tail to the right. This shows that a lot of simulation runs satisfy the condition of the combined stationarity test from the beginning. This happens because the initial queue length of zero customers is the system state with the highest probability during steady state.

Figure 5.24 shows again the distributions for the M/M/1 queue, but here $\rho = 0.8$ and there are one hundred initial customers, compare with Table 5.14. The distributions of the estimated truncation points of both methods look similar again. Both distributions have a long right tail and a short tail to the left. It is unlikely that either test selects the truncation point at observation index zero. This shows that the high initial state introduces a transient behaviour which is easier to recognise for the combined stationarity test than in the previous example.

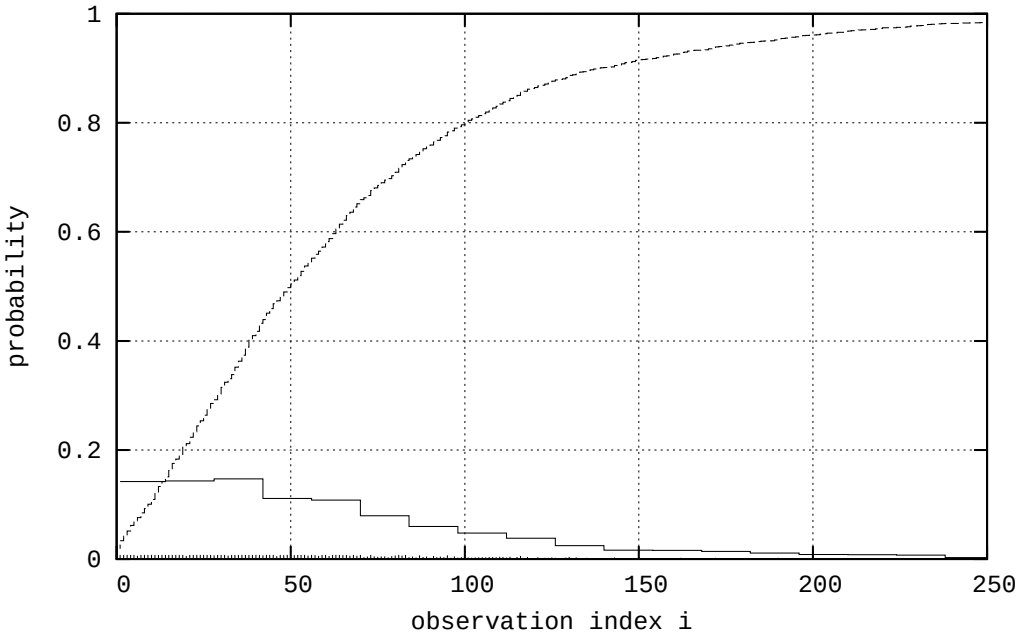


(a) combined stationarity test

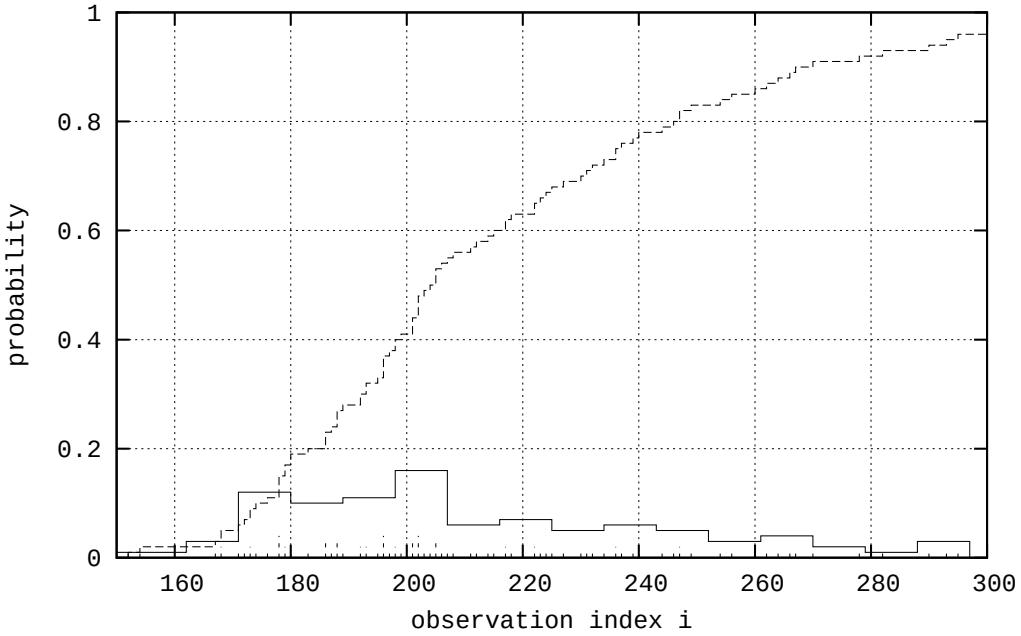


(b) homogeneity-based truncation-point estimator

Figure 5.20: Histogram and empirical CDF of the estimated truncation points for the parabola displacement, see Equation (5.32) and compare with Table 5.6.

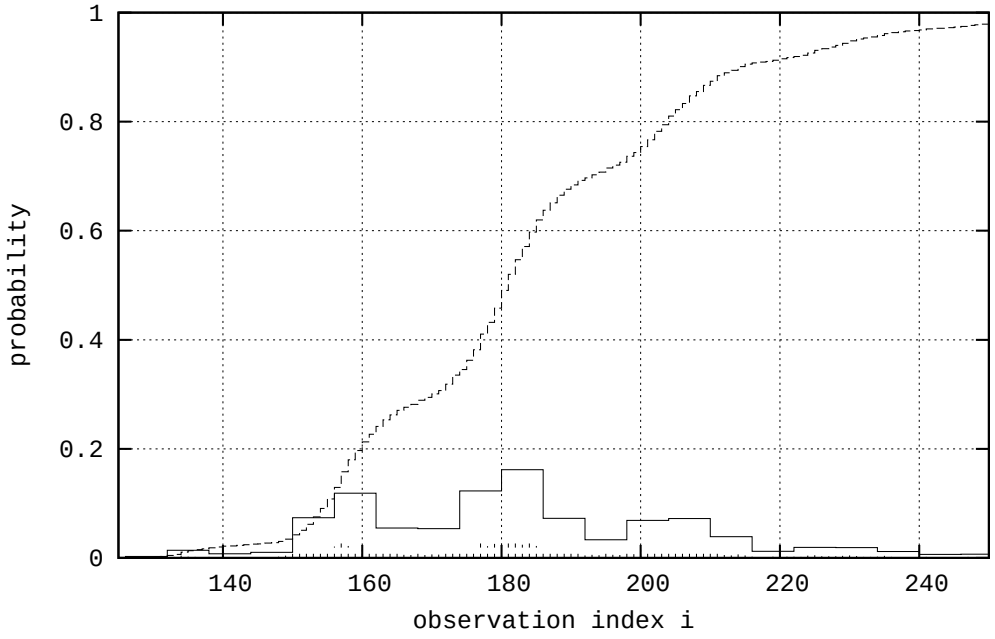


(a) combined stationarity test

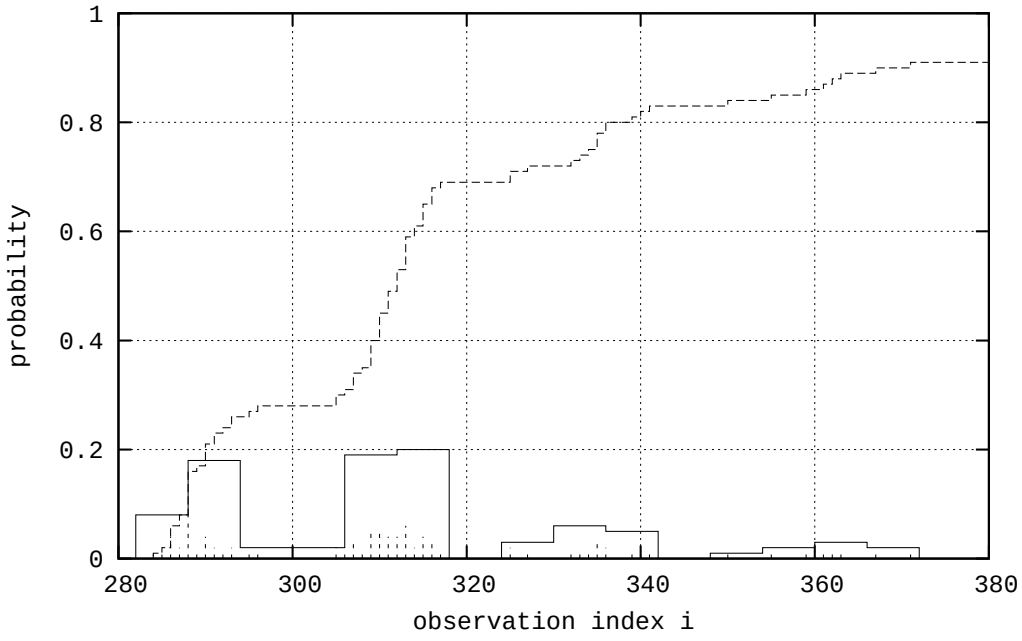


(b) homogeneity-based truncation-point estimator

Figure 5.21: Histogram and empirical CDF of the estimated truncation points for a geometrical ARMA$(5, 5)$ process, see Equation (5.33) with $\Upsilon_i^{(5)} = 0$ for $i \leq 0$ and compare with Table 5.7.

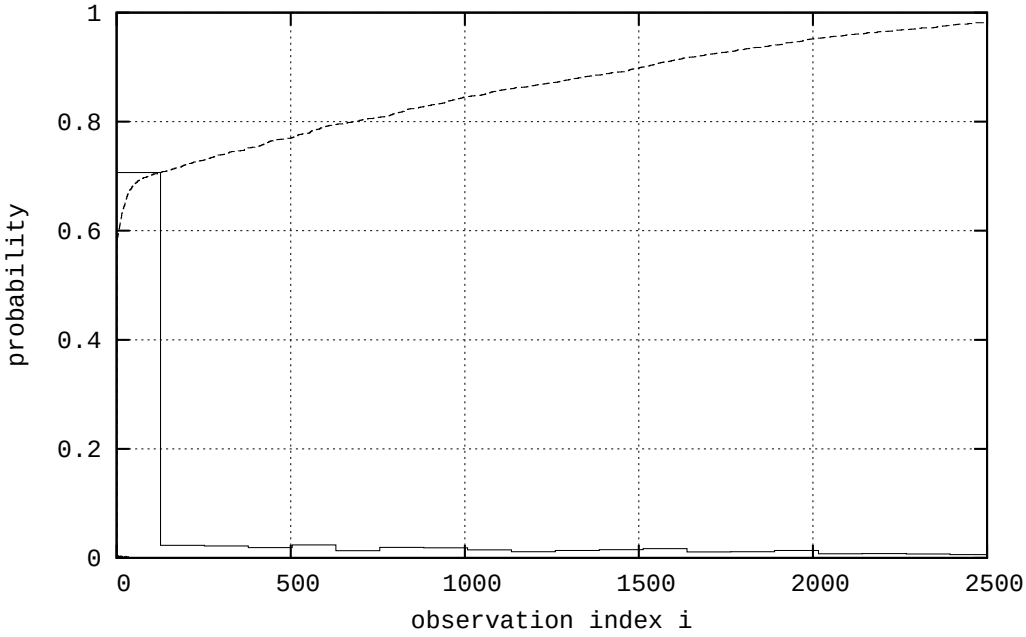


(a) combined stationarity test

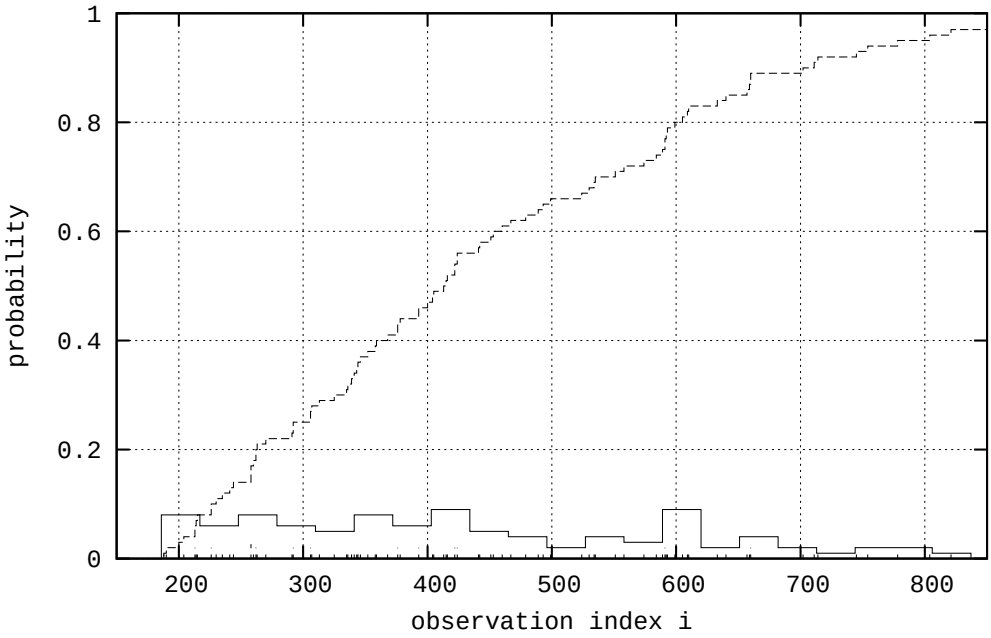


(b) homogeneity-based truncation-point estimator

Figure 5.22: Histogram and empirical CDF of the estimated truncation points for the process governed by a damped vibration, see Equation (5.34) and compare with Table 5.8.

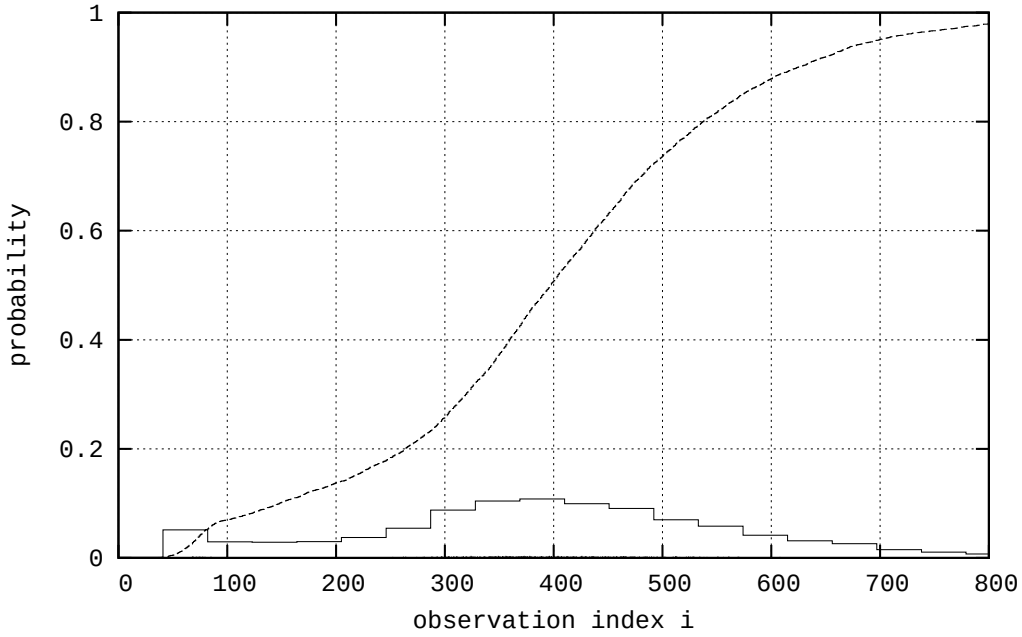


(a) combined stationarity test

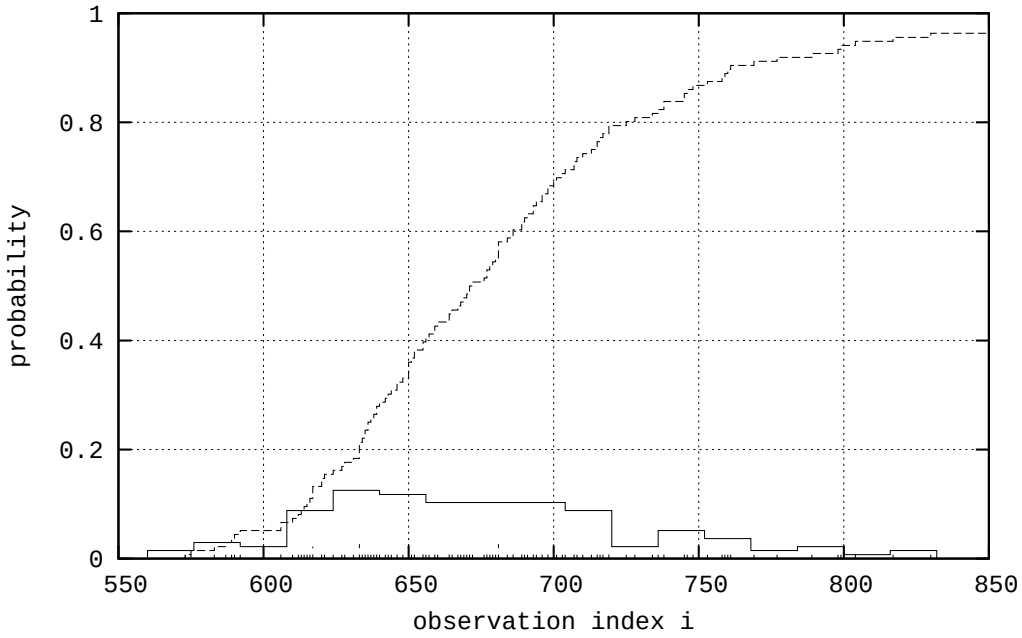


(b) homogeneity-based truncation-point estimator

Figure 5.23: Histogram and empirical CDF of the estimated truncation points for the M/M/1 queue with $\rho = 0.95$ and no initial customers, compare with Table 5.12.



(a) combined stationarity test



(b) homogeneity-based truncation-point estimator

Figure 5.24: Histogram and empirical CDF of the estimated truncation points for the M/M/1 queue with $\rho = 0.8$ and one hundred initial customers, compare with Table 5.14.

5.6.6 Interpretation

The crossing of the mean rule is just a simple heuristic. No real null hypothesis is tested and, therefore, no significance level can be given. It is easy to find processes where this method fails.

As we have seen in the previous sections, the parameters of the combined stationarity test have to be chosen for every model separately. Especially the batch size m is problematic. If m is too small the batch means will still be correlated and the estimation of the variance is doubtful. If m is too big the transient phase may be just a small part of the first window and the combined stationarity test may overlook the transient behaviour. The settings we chose for m in our examples led to good simulation results. We believe that it might be possible to choose even better settings for m, however, this would not change the general outcome of the experiments. Another problem with the combined stationarity test is that it assumes the variance to be smaller during the steady state phase than during the transient phase. We gave examples that violate this assumption. If the analyst does not know whether the variance is increasing or decreasing over time a two-sided test should be used. This reduces the power of the test because each direction has to be tested at a lower significance level of $1 - \frac{\alpha}{2}$.

In general the homogeneity-based truncation-point estimator delivers truncation points, which are deepest within the steady state phase. This is because the homogeneity test checks equality of the distribution. In contrast to this the crossing of the mean rule targets at a stable mean and the combined stationarity test checks a stable variance. These are just necessary conditions for steady state in terms of the probability distribution and are included in the homogeneity test.

Despite all these positive results of the homogeneity-based truncation-point estimator, i.e. Listing 5.1, we can see that the estimated truncation points are still not deep enough in the steady state phase. This problem is caused by the algorithmic approach which aims at a precise estimation of the theoretically best

truncation point. The results of Listing 5.1 are often very close to the theoretically best truncation point, but still too short. This problem can be solved by the more wasteful algorithmic approach of Listing 5.3. This is discussed in the next section.

5.6.7 Time and Memory Efficient Algorithm

The experiments done in the previous sections aim at the estimation of a precise truncation point. We have seen that the initialisation bias can still remain after deletion, if the estimated truncation point is smaller than the theoretically best truncation point. For reliable estimators in subsequent analysis, which assume identically distributed data, it is not important to know the truncation point very precise, but to choose a truncation point which is beyond the theoretically best truncation point. To avoid unnecessary wasteful methods the estimated truncation point should still be close to the theoretical value.

In the software tool Akaroa2 a truncation point detection method is implemented which is a combination of the crossing of the mean rule and Schruben's test, see [118-SST83]. Thus it covers mainly Equation (5.3), and would also cover Equation (5.4) in some selected examples. Mixing two different strategies hides their weaknesses, this is why this combination was not used before. Note, Schruben's test of [118-SST83] was replaced in [64-GSS94] by a newer version, which is used in experiments of previous sections. We will denote the final estimate of this method by $\bar{l}_E$. During the heuristic phase of the method a window size is estimated by the crossing of the mean rule. This window size is used in Schruben's test to verify that the data is identically distributed and there is no trend. If the test fails, the window is shifted by its complete size. The estimated truncation point is set to be the end of the window, which contains the data that is proved to be without a trend. For more details see e.g. [55-Gho04]. The combination of two different truncation point detection methods makes this approach more powerful than most other approaches which are based on just one criterion.

The time and memory efficient version of the homogeneity-based truncation-point estimator, i.e. Listing 5.3, operates by doubling the actual candidate for a truncation point, if one of the homogeneity test fails. Furthermore, the random sample at the actual candidate is compared only with a selection of subsequent random samples. As in the truncation point detection method of Akaroa2, the final truncation point is set to be at the end of the so far processed data.

Both methods do not search continuously for a truncation point. Therefore, it is not possible to draw the empirical distribution function of their estimates or derive any other measure than the average over all simulation experiments. In the previous section we demonstrated that especially the queueing models produce output processes which are difficult to analyse. We repeat those experiments with the more promising algorithmic approach and report average truncation points.

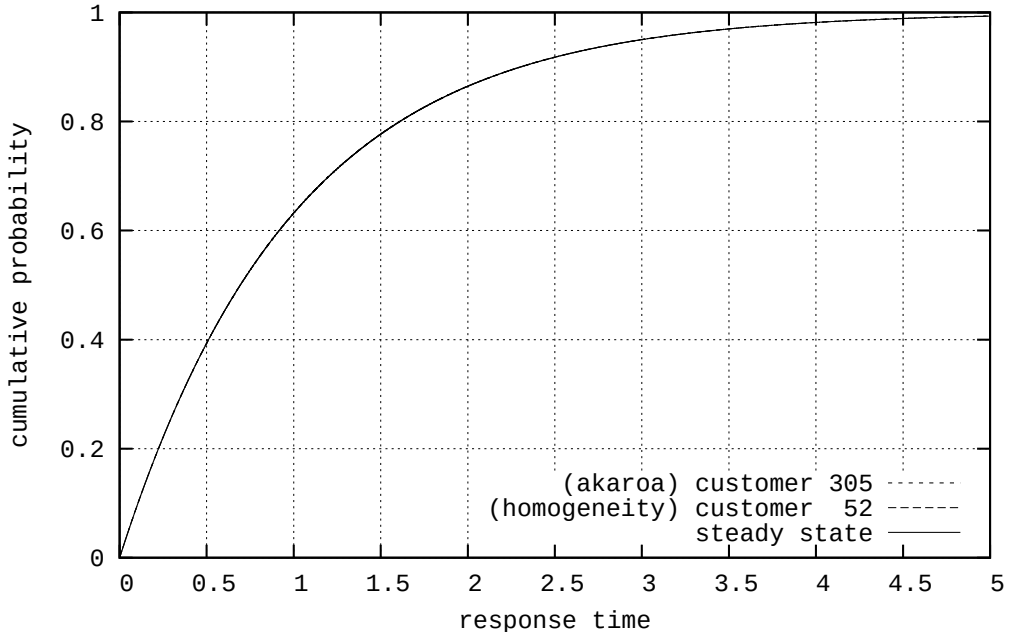
For the M/M/1 queue we chose an interarrival rate $\lambda = 1$ and a service rate $\mu = \{\frac{1}{0.5}, \frac{1}{0.95}\}$, leading to the traffic intensity $\rho = \{0.5, 0.95\}$. The parameters of the M/E_2/1 queue are chosen similarly. The interarrival rate is $\lambda = 1$ and the service rate of the Erlang distribution is $\mu = \{\frac{1}{0.5}, \frac{1}{0.95}\}$ with shape 2. The traffic intensity is here also $\rho = \{0.5, 0.95\}$. In both examples we observed the system's response time R_i of the ith customer.

The average of all estimated truncation points of every method and every queueing model at all traffic intensities is shown in Table 5.15. We can see, that the average truncation point of the method implemented in Akaroa2 is larger than the result of the homogeneity-based truncation-point estimator for $\rho = 0.5$

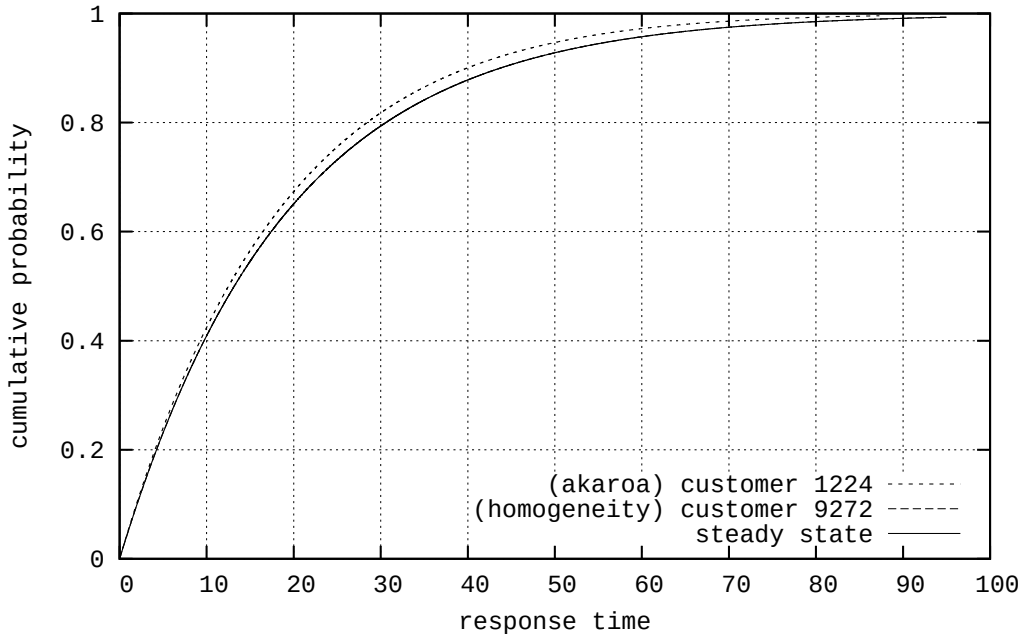
	Akaroa2	homogeneity
M/M/1 0.50	305.3 ± 0.3	52 ± 3
M/M/1 0.95	1224 ± 12	9272 ± 545
M/E_2/1 0.50	283.2 ± 0.2	54 ± 4
M/E_2/1 0.95	1163 ± 4	8378 ± 556

Table 5.15: Average of all estimated truncation points.

in both queueing models. For $\rho = 0.95$ the situation is reversal. In first place the distribution function at the estimated truncation point should be identical to the steady state distribution and in second place it should be small to avoid the deletion of too much data. In Figure 5.25(a) and Figure 5.26(a) we can see that the distribution function at the estimated truncation points of both methods with $\rho = 0.5$ is indistinguishable from the steady state distribution. The average estimate of the homogeneity-based truncation-point estimator is smaller than the average estimate of the method implemented in Akaroa2. The method implemented in Akaroa2 is here unnecessary wasteful because it deletes too much data. For $\rho = 0.95$ the situation is different, the homogeneity-based truncation-point estimator deletes more data. Figure 5.25(b) and Figure 5.26(b) show that this is necessary. The graph of the steady state distribution covers only the graph of the homogeneity-based truncation-point estimator. The distribution function at the truncation points estimated by Akaroa2 are still different to the steady state distribution. We can predict, that the coverage of subsequent estimators will be smaller for $\rho = 0.95$ than for $\rho = 0.5$ due to inaccurate deletion of the initial transient phase. In general, the homogeneity-based truncation-point estimator with the appropriate algorithmic approach of Listing 5.3 is more precise, less wasteful and a bias on subsequent steady state estimators can almost be excluded. It implements Equation (5.1), which covers Equation (5.3) and Equation (5.4), as discussed in Section 5.1.

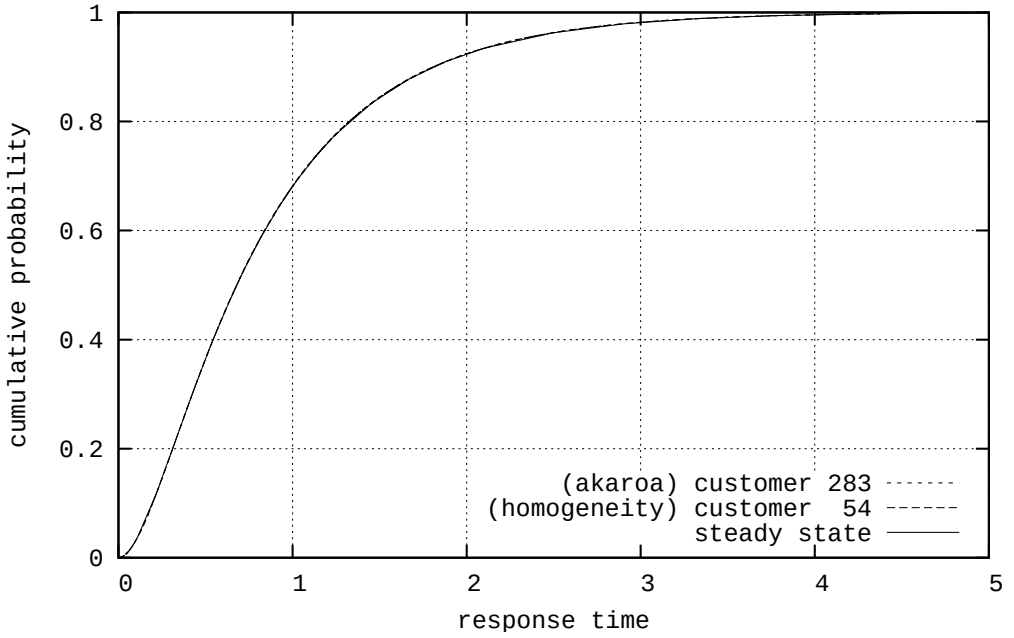


(a) traffic load $\rho = 0.5$

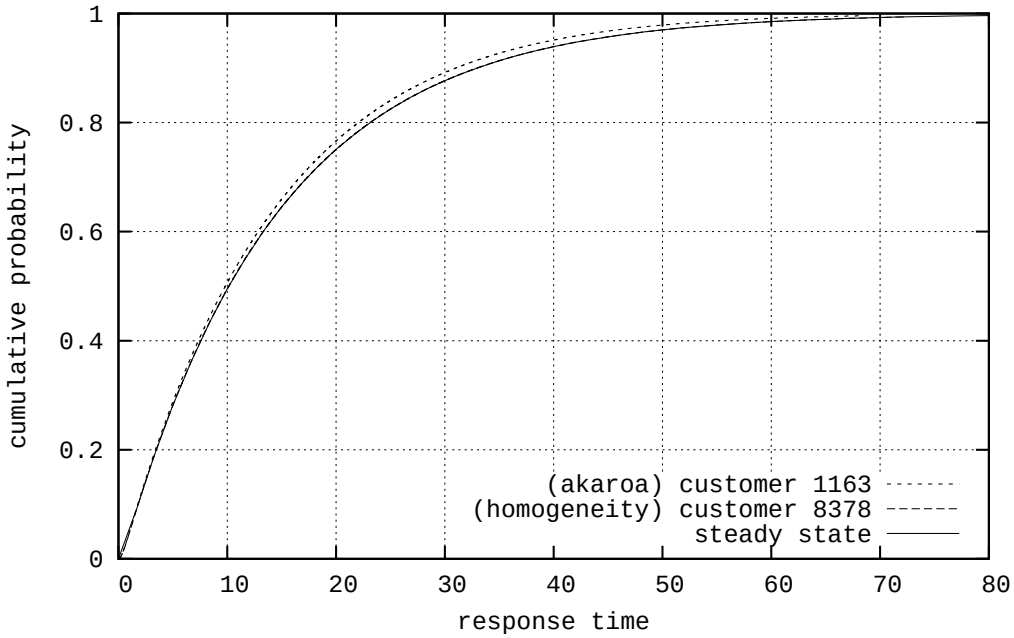


(b) traffic load $\rho = 0.95$

Figure 5.25: $F_{R_i}(x)$ at the average truncation points ($\bar{l}_E = \{305, 1224\}$; $\bar{l}_F = \{52, 9272\}$) of the M/M/1 queue compared with the steady state distribution $F_{R_\infty}(x)$ at different traffic loads ρ.



(a) traffic load $\rho = 0.5$



(b) traffic load $\rho = 0.95$

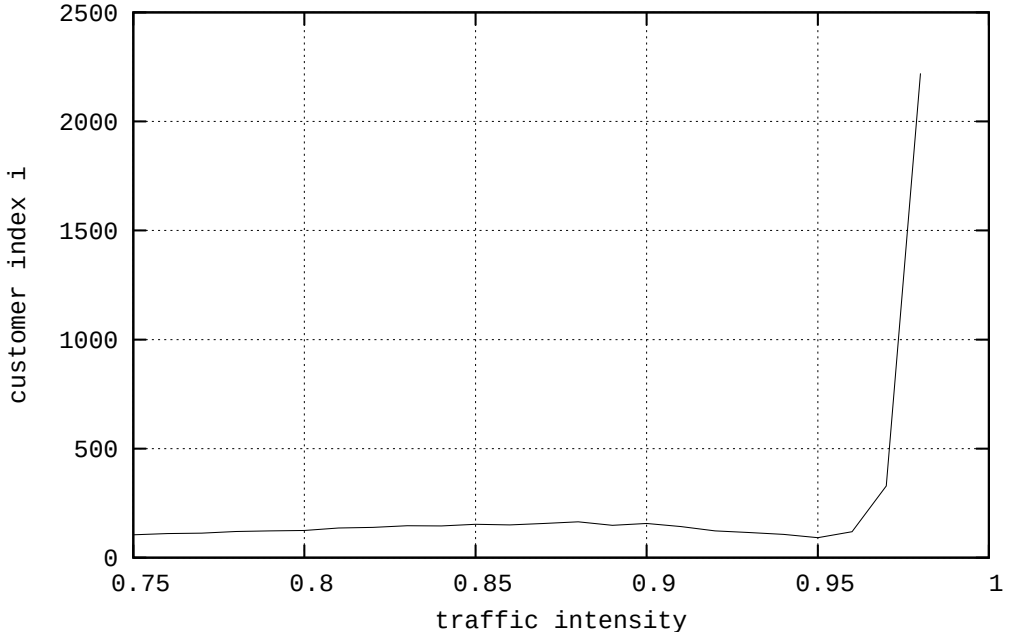
Figure 5.26: $F_{R_i}(x)$ at the average truncation points ($\bar{l}_E = \{283, 1163\}$; $\bar{l}_F = \{54, 8378\}$) of the M/E_2/1 queue compared with the steady state distribution $F_{R_\infty}(x)$ at different traffic loads ρ.

5.6.8 Versions of Homogeneity-Based Estimators

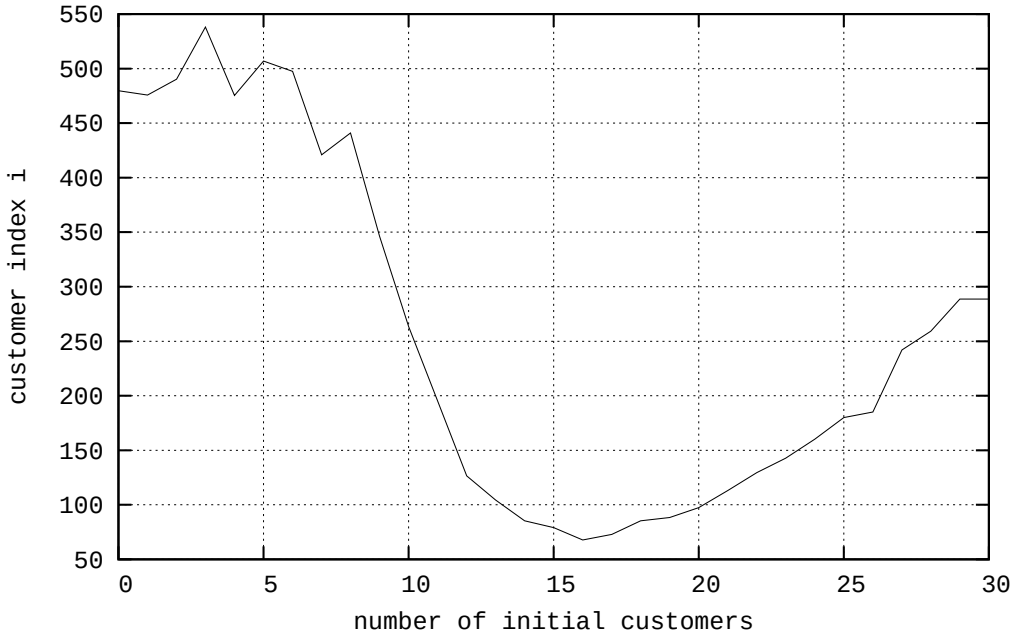
In previous sections we stated that the algorithmic approach of Listing 5.3 detects truncation points deeper within the steady state phase than the other algorithmic approaches of Listing 5.1 and Listing 5.2. Here, we will prove this experimentally for the M/M/1 queue. The results of Listing 5.1 are shown in Figure 5.27, the results of Listing 5.2 are shown in Figure 5.28 and the results of Listing 5.3 are shown in Figure 5.29. The curves in Figure 5.27(a), Figure 5.28(a) and Figure 5.29(a) show an M/M/1 queue with 19 initial customers and we varied $0.75 \leq \rho \leq 0.98$ by varying μ and keeping $\lambda = 1$ constant. For the curves in Figure 5.27(b), Figure 5.28(b) and Figure 5.29(b) we varied the number of initial customers $0 \leq N_0 \leq 30$ and kept $\rho = 0.95$ constant with $\lambda = 1$. All curves show mean values of l_F based on one hundred simulation experiments with $p = 100$.

We can see that the general form of the curves is similar no matter which algorithmic approach is used. However, it is clearly evident that the estimates of l_F of Listing 5.3 are greater than the estimates of both other methods. This supports our previous assumption that Listing 5.3 detects truncation points deeper within the steady state phase than the other algorithmic approaches.

Quite interesting is the location of the minimum of the depicted curves. In Figure 5.27(a), Figure 5.28(a) and Figure 5.29(a) the minimum is located at $\rho = 0.95$. Because we used $N_0 = 19$ initial customers this conforms to theory, $\mathrm{E}[N_\infty] = \frac{\rho}{1-\rho} = \frac{0.95}{1-0.95} = 19$. In Figure 5.27(b) and Figure 5.28(b) we can see that the minimum is located at points smaller than $N_0 = 19$. This is contrary to what we found out in Section 5.2. However, the minimum in Figure 5.29(b) is exactly at $N_0 = 19$, which conforms closer to theory.

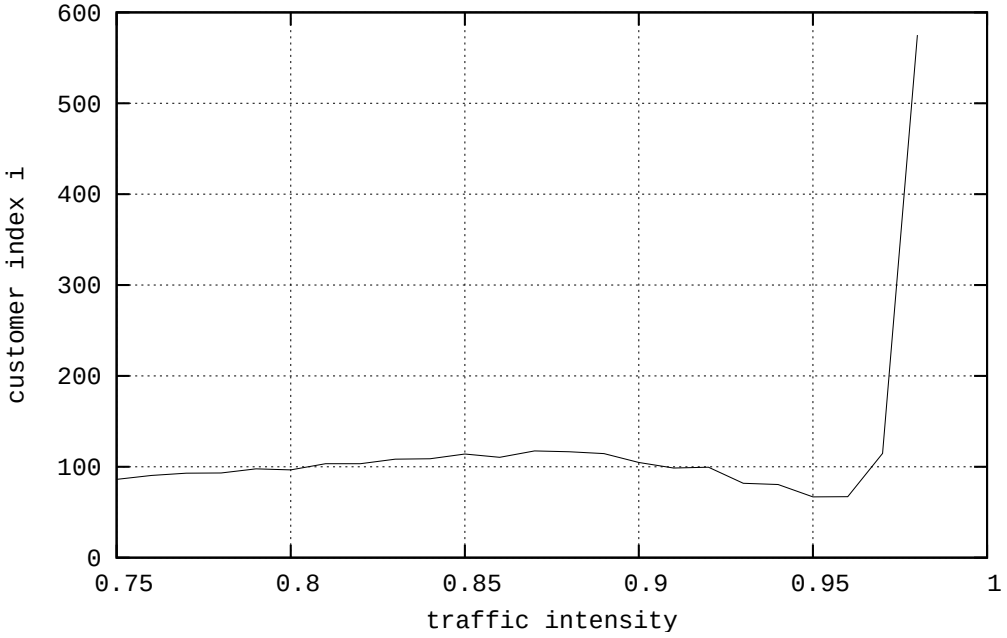


(a) Mean of l_F in dependence of $0.75 \leq \rho \leq 0.98$, $N_0 = 19$.

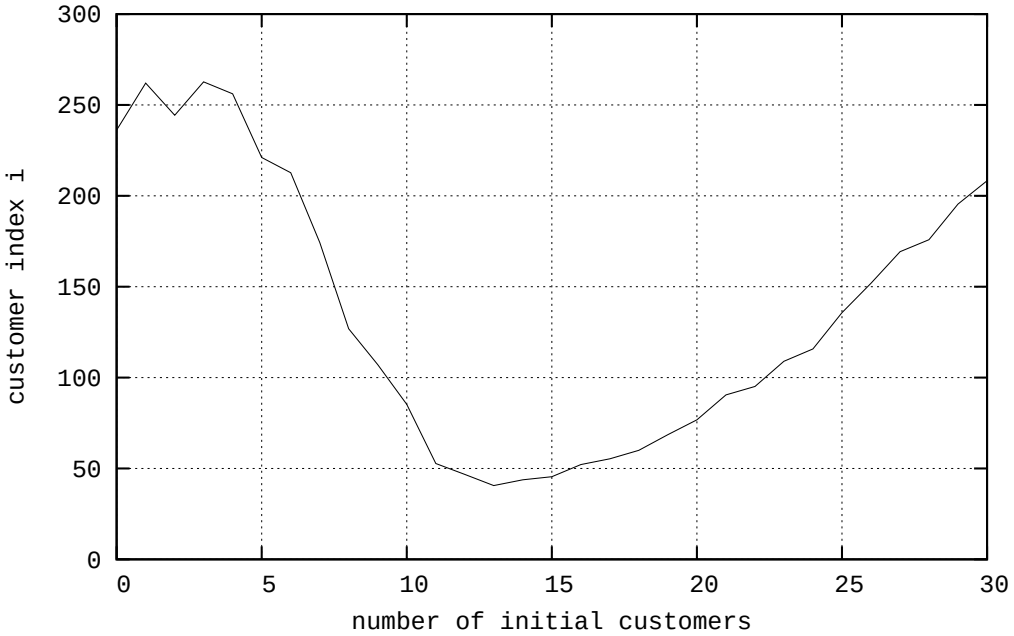


(b) Mean of l_F in dependence of $0 \leq N_0 \leq 30$, $\rho = 0.95$.

Figure 5.27: Results of Listing 5.1; mean of all estimated truncation points l_F of an M/M/1 queue with traffic intensity ρ and N_0 initial customers, based on a hundred experiments with $p = 100$.

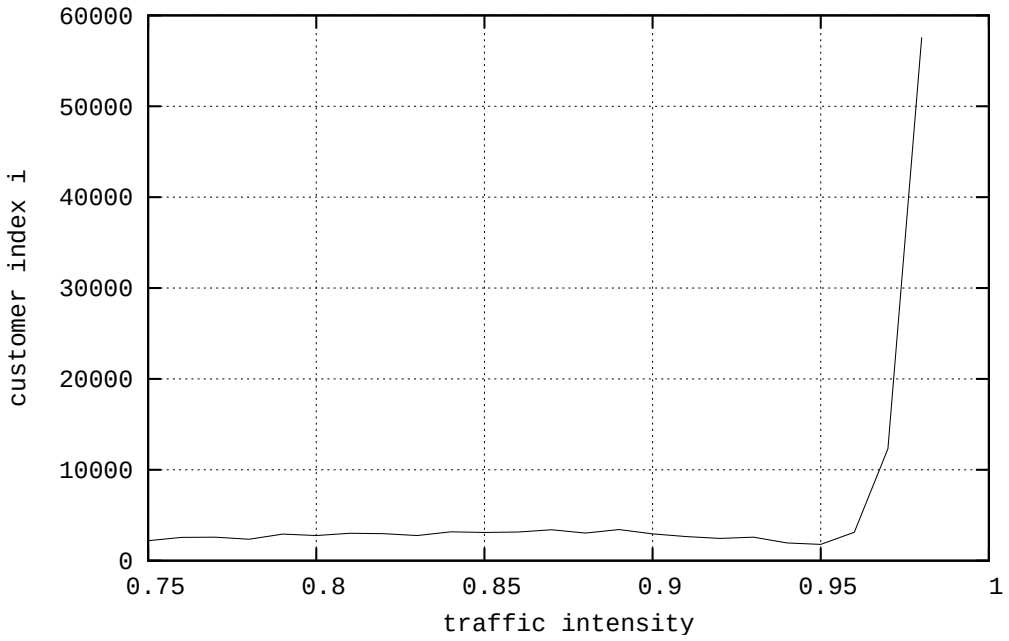


(a) Mean of l_F in dependence of $0.75 \leq \rho \leq 0.98$, $N_0 = 19$.

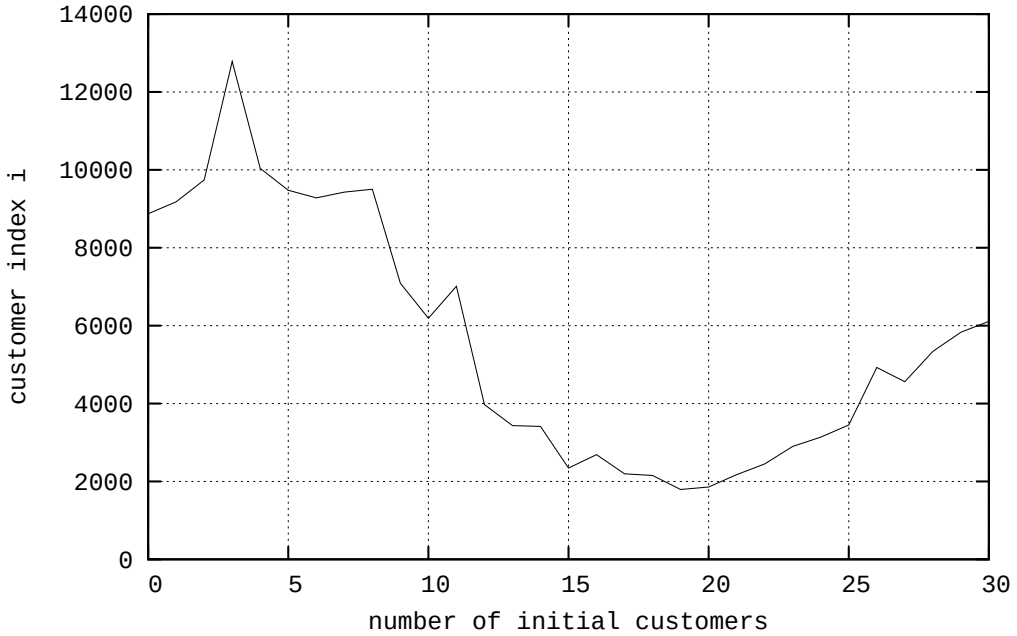


(b) Mean of l_F in dependence of $0 \leq N_0 \leq 30$, $\rho = 0.95$.

Figure 5.28: Results of Listing 5.2; mean of all estimated truncation points l_F of an M/M/1 queue with traffic intensity ρ and N_0 initial customers, based on a hundred experiments with $p = 100$.



(a) Mean of l_F in dependence of $0.75 \leq \rho \leq 0.98$, $N_0 = 19$.



(b) Mean of l_F in dependence of $0 \leq N_0 \leq 30$, $\rho = 0.95$.

Figure 5.29: Results of Listing 5.3; mean of all estimated truncation points l_F of an M/M/1 queue with traffic intensity ρ and N_0 initial customers, based on a hundred experiments with $p = 100$.

5.7 Limits and Conclusion

In the previous section we demonstrated that the use of homogeneity tests on the output of multiple independent parallel replications opens a new class of truncation point detection methods, which are based on the convergence of the probability distribution towards its steady state behaviour. This class covers the truncation point detection in a broader sense, i.e. it implements Equation (5.1). Apart from [9-AG06], practically all other methods cover steady state of the mean (see Equation (5.3)) or steady state of the variance (see Equation (5.4)).

In Section 5.2 we discussed the transient behaviour of an M/M/1 queue. Even though the M/M/1 queue is quite simple, in comparison to other simulation models, its transient behaviour can be quite complex. In this example it seems to be necessary to use Equation (5.1) to detect the steady state phase. Simplifying assumptions, like Equation (5.8), do not hold for the M/M/1 queue.

The three different algorithmic approaches of Section 5.4 have a slightly different focus. The approaches of Listing 5.1 and Listing 5.2 aim at the estimation of l close to the theoretically best choice, whereas the approach of Listing 5.3 selects an l beyond the best choice. The first two versions are appropriate if the creation of output data is very time consuming. The third version is more wasteful and should be used if the creation of output data is not very time consuming.

However, there are also some limitations for the first two versions. As we already pointed out, the algorithmic approaches of Listing 5.1 and Listing 5.2 tend to estimate l smaller than the theoretically best choice. This may cause problems for subsequent analysis. An estimator, that is assuming identically distributed observations, would still be biased. Note, that this bias is much smaller than the bias of non truncated data. To obtain estimates of l which are greater than the best choice, we introduced the algorithmic approach of Listing 5.3. In general a bias on subsequent estimators can be excluded. Its worst case time complexity is the

best and its storage requirement is small and constant. Therefore, we recommend the approach of Listing 5.3.

All homogeneity tests, which are discussed in Section 5.3, operate on a selected significance level. This is the reason why true equality in distribution cannot be tested. If the difference in distribution of consecutive random samples is very small, it might be overlooked by the homogeneity tests, compare Equation (5.2). In general, the difference in distribution is growing during the transient period if the distance in observation time of two random samples is growing. Thus, the ratio parameter r can prevent the homogeneity tests from overlooking a small difference. The automated detection of r relies on searching for the first two random samples, which have a noticeable difference in distribution. However, because the rate of the convergence of the probability distribution might change over time the general problem remains. Large errors can be avoided by using a boundary value $r \geq r_{min}$, as it is suggested in Section 5.5.

The outcome of homogeneity tests depends on the size of the compared random samples. In our approach a random sample is established by taking one observation from each replication. Thus, our sample size p is given by the number of parallel replications. In Section 5.5 we showed that $p \geq 30$ should be used and that a good choice is $p = 100$. A hundred parallel replications is quite large. However, we have to keep in mind that we do not need p computers or processors to execute p parallel replications. Depending on the memory requirements of the simulated model, all replications could even be executed on only one computer. This might even be advisable to avoid unnecessary network traffic.

The comparison is done with methods which have a different focus of analysis. As stated previously, the application area of both, Schruben's methods in its various versions and the crossing of the mean rule, is mean value analysis. These methods have been applied in automated simulation analysis, for example in Akaroa2. There are no other methods available which follow exactly our aim

of detecting steady state in sense of the underlying distribution and which are well enough documented for implementation. Furthermore, our method uses multiple replications. Here, the use of synchronous data collection enables new techniques of online analysis of simulation output data.

In this chapter we focus entirely on reducing the bias caused by the initial state. We reject using different initial states because this approach is not possible in automated simulation without further knowledge of the steady state distribution, as discussed in the introduction of this chapter. Choosing the initial state of each replication empirically is a possible alternative. However, this alternative does not guarantee that the initial transient gets shorter in comparison to choosing the same initial state for all replications. In mean value analysis of queueing models the system state "empty & idle" is selected as a save initial state, because it is known to lead to not an excessively long initial transient, see [79-KL85]. There are initial states other than "empty & idle" state which can be associated with shorter initial transients, however, the possibility of choosing an initial state that leads to a long initial transient is high if one simulates a system of unknown dynamics. For the same reason it is advisable to choose a save initial state for all replications when analysing quantiles. Alternative strategies of initialisation do not guarantee a shorter initial transient.

Because of the occurrence of ties in random samples of discrete random variables the test statistic of homogeneity tests has to be adjusted. Thus, we have to distinguish between stochastic output process with a discrete or a continuous range. Our implementation has only been tested for the continuous case.

In Section 5.6 we demonstrated on a variety of different output processes, that this new class of truncation point detection methods is more powerful than methods of other classes. The estimated truncation point l is in general deeper within the steady state phase and it works fine for a wider range of models. The use of multiple independent parallel replications offers not only a speed up in the

creation of output data, furthermore, it enables the implementation of a new class of truncation point detection methods.

Chapter 6

Quantiles in Steady State

In Chapter 2 we outlined the basic mathematics of quantile estimation and the situation of simulation output analysis is discussed as application area. Here, we would like to develop methods for automated and sequential analysis to estimate a set of quantiles. This set of quantiles can be used to depict the underlying CDF. Parts of the discussion and results of this chapter are published in [38-Eic06] and [44-EMP07b].

First, we will check the suitability for automated analysis of the most promising quantile estimator. This is done in Section 6.1. In Section 6.2 we derive further quantile estimators by applying estimation techniques from mean value analysis to the area of quantile analysis. Their performance is analysed and discussed. A new quantile estimator, which is based on estimation techniques for independent and identically distributed random samples, is introduced in Section 6.3. Its performance is also analysed. In Section 6.4 experimental studies are done with different classes of models to test different properties of the estimators. This chapter ends with some conclusions in Section 6.5.

6.1 Distribution Estimated by Several Quantiles

A quantile $x_q = F_X^{-1}(q)$ is the point (x_q, q) of the CDF of X. Several estimated quantiles with carefully selected q_1, q_2, ... can give an impression of $F_X(x)$ and

can be used to estimate $F_X(x)$.

Probably the best and most studied method for the calculation of several quantiles is Raatikainen's method, see Section 2.4.2. Let us briefly look at this method. This method seeks to estimate the probability of predefined intervals $\mathcal{C}_k$. By defining $\mathcal{C}_k = [-\infty, x_k]$ the probability $q_k = F_X(x_k)$ is estimated. The desired halfwidth δ_k (see Equation (2.25)) and the combined confidence level $1 - \alpha$ (see Equation (2.28)) have to be specified by the analyst. Furthermore, the number of batches used for spectral analysis has to be set. The standard setting is 512 batches, as this value is chosen in [110-Raa95] for experiments with an M/M/1 queue.

Table 6.1 lists the results of our coverage analysis of Raatikainen's method obtained for $m = 25$ estimated quantiles and $\alpha = 0.05$. Reported are mean values of all quantiles. The column *halfwidth* shows the halfwidth of the 95% confidence interval of the estimated coverage. The column *coverage* shows the probability that all $F_X(x_k)$ are within the estimated confidence intervals. Because

model	coverage	halfwidth	runs
normal process	≈ 1	-	10^4
uniform process	0.9998	< 0.0003	10^4
exponential process	≈ 1	-	10^4
geometrical ARMA$(1,1)$	0.9994	< 0.0005	10^4
geometrical ARMA$(2,2)$	0.9983	< 0.0009	10^4
M/M/1/∞ $\rho = 0.5$	0.9969	< 0.0011	10^4
M/M/1/∞ $\rho = 0.75$	0.9953	< 0.0014	10^4
M/M/1/∞ $\rho = 0.9$	0.9962	< 0.0013	10^4
M/E_2/1/∞ $\rho = 0.5$	0.9999	< 0.0002	10^4
M/E_2/1/∞ $\rho = 0.75$	≈ 1	-	10^4
M/E_2/1/∞ $\rho = 0.9$	≈ 1	-	10^4
M/H_2/1/∞ $\rho = 0.5$	0.9995	< 0.0005	10^4
M/H_2/1/∞ $\rho = 0.75$	0.9992	< 0.0006	10^4
M/H_2/1/∞ $\rho = 0.9$	0.9981	< 0.0009	10^4

Table 6.1: Mean coverage of all quantile estimates of Raatikainen's method, see Section 2.4.2, where the expected coverage is 0.95.

we set $\alpha = 0.05$ the expected coverage is 0.95. We can see that the estimated coverage is higher than the assumed confidence level and close to one. We based or results on a time consuming number of 10^4 independent results, see column *runs*. A fixed number of independent repetitions is used e.g. in [85-LK00]. An estimated coverage close to one shows that Raatikainen's point estimate is very precise for $\delta_k = \arcsin(0.05)$ (the value is suggested in [110-Raa95]). However, in our example the interval estimate is too conservative. This might be due to Bonferroni's inequality, see Equation (2.28). It gives an upper bound, which is possibly quite conservative and too strict.

We have found that the parameterisation of Raatikainen's method is difficult in general. The analyst has to choose adequate δ_k which are probabilities. As well as this, the analyst has to choose x_1, x_2, ..., x_m which are in the domain of the measure itself. Furthermore, the domain of $F_X(x)$ must be known in order to choose a good set of x_1, x_2, ..., x_m. If the domain of $F_X(x)$ is unbounded, i.e. $-\infty \leq x \leq \infty$, an even deeper prior knowledge is needed to place all x_1, x_2, ..., x_m in the most interesting area. We can see, that Raatikainen's method is difficult for automated analysis. It is impossible to set the x_k in critical parts of $F_X(x)$ for an unknown and arbitrary distribution.

The next problem for the analyst is to decide the number m of estimated quantiles. How many quantiles are needed to obtain a reasonable estimate of the curve of $F_X(x)$? It is not a good idea to choose m as large as possible because of Bonferroni's inequality in Equation (2.28). $5 \leq m \leq 25$ is recommended in [110-Raa95]

Furthermore, all estimates q_1, q_2, ..., q_m are correlated, because they are calculated from the same simulation output process. A higher value of m leads to smaller distances between neighbouring x_k and the predefined intervals $\mathcal{C}_k = [-\infty, x_k]$ will be less disjoint. Equation (2.23) describes the correlation between q_1, q_2, ..., q_m. Correlation is high if values of x_k are located closely together. This is the reason for the use of Bonferroni's inequality in Equation (2.28). However,

Raatikainen's method cannot suggest the optimal value for m, which should also take the choice of δ_k into account.

We can conclude that Raatikainen's method and other methods described in Section 2.4 are not the best choice for automated analysis of several quantiles, despite their good statistical properties. In the next sections we will derive methods for the estimation of several quantiles, which are based on multiple independent replications. They are applicable in sequential and automated analysis and have good statistical properties.

6.2 Batch Means and Spectral Analysis for Order Statistics

Let us assume that having applied the method described in Chapter 5 the remaining output process $\{X_i\}_{l_F=i}^{\infty}$ is in its steady state phase, where l_F is the truncation point as defined by Equation (5.1). Thus, we can assume that all random variables X_i of the output process $\{X_i\}_{i=l_F}^{\infty}$ have the same marginal distribution, i.e. Equation (2.5) holds, and the data of the beginning $l_F - 1$ observation indexes are truncated.

Using p multiple replications, as in Chapter 5, we obtain the observations $x_{j,i}$, where $1 \leq j \leq p$ is the replication index and $l_F \leq i < \infty$ is the observation index. The independence of all replications implies that the observations $\{x_{j,i}\}_{j=1}^{p}$ are independent of each other. This is valid for all observation indexes i. Therefore, for a fixed observation index the statistical methods which are discussed in Section 2.2, are directly applicable to the observations $\{x_{j,i}\}_{j=1}^{p}$.

Here, the definition of the population quantile, see Equation (2.12), has to be extended by adding the observation index i:

$$x_{q,i} = F_{X_i}^{-1}(q) = \inf\{x | F_{X_i}(x) \geq q\}. \tag{6.1}$$

Let $\{y_{j,i}\}_{j=1}^{p}$ be the ordered values of $\{x_{j,i}\}_{j=1}^{p}$ and let $\{y_{j,i}\}_{i=l_F}^{\infty}$ be a realisation of

the stochastic process $\{Y_{j,i}\}_{i=l_F}^{\infty}$. In this sense $Y_{j,i}$ represents the jth order statistic at observation index i.

For a given probability q the quantile $F_{X_i}^{-1}(q)$ can be estimated by the sample quantile

$$\hat{x}_{q,i} = y_{\lfloor pq+1 \rfloor, i}, \tag{6.2}$$

(compare with Equation (2.14)). The homogeneity test of Section 5.3 ensures that the difference in distribution between the remaining X_i, where $i \geq l_F$, is negligible. In consequence, for fixed j all $\{y_{j,i}\}_{i=l_F}^{\infty}$ describe a quantile of the steady state probability distribution $F_X(x)$, so that the average value can be calculated by

$$\hat{x}_q = \frac{1}{n - l_F + 1} \sum_{i=l_F}^{n} \hat{x}_{q,i}, \tag{6.3}$$

which is a point estimate of $F_X^{-1}(q)$ and where n is the current simulation horizon.

Theorem 6.2.1 *$\hat{x}_q$ is an asymptotically unbiased estimator of $F_X^{-1}(q)$ for large p, where $n - l + 1$ is the number of order statistics used for calculation.*

Proof The expected value of Equation (6.3) is

$$\begin{aligned} \mathrm{E}\left[\hat{x}_q\right] &= \frac{1}{n - l_F + 1} \sum_{i=l_F}^{n} \mathrm{E}\left[\hat{x}_{q,i}\right] \\ &= \frac{1}{n - l_F + 1} \sum_{i=l_F}^{n} \mathrm{E}\left[y_{\lfloor pq+1 \rfloor, i}\right]. \end{aligned} \tag{6.4}$$

$\mathrm{E}\left[y_{\lfloor pq+1 \rfloor, i}\right] = F_{X_i}^{-1}(q)$ holds for large values of p, see [34-Dav70] or [24-Che02]. Furthermore, all X_{l_F}, X_{l_F+1}, ... are assumed to be identically distributed, i.e. $\forall i : F_{X_i}(x) = F_X(x)$. Equation (6.4) evaluates to

$$\begin{aligned} \mathrm{E}\left[\hat{x}_q\right] &= \frac{1}{n - l_F + 1} \sum_{i=l_F}^{n} F_{X_i}^{-1}(q) \\ &= \frac{1}{n - l_F + 1} \sum_{i=l_F}^{n} F_X^{-1}(q) \\ &= F_X^{-1}(q). \end{aligned} \tag{6.5}$$

The estimator $\hat{x}_q$ is asymptotically unbiased, i.e. $\mathrm{E}\left[\hat{x}_q\right] - F_X^{-1}(q) = 0$, because Equation (6.5) holds for large p and $i \geq l_F$. ∎

Our aim is to estimate not only one quantile at a given probability q but to estimate several quantiles of $F_X(x)$. Here, a natural approach is to calculate those quantiles which are represented by the jth order statistics $Y_{j,i}$ at observation index i. The quantile representation of the jth order statistic depends on the form of the distribution $F_X(x)$ (see Equation (2.15), Equation (2.16) and Equation (2.17)). The probability q_j, associated with the quantile x_{q_j}, which is represented by the jth order statistic, can be estimated by

$$\hat{q}_j = \begin{cases} \frac{j}{p+1} & \text{(unknown / uniform case),} \\ \frac{j}{p+\frac{1}{2}} & \text{(exponential case),} \\ \frac{j-\frac{1}{2}}{p} & \text{(normal case),} \end{cases} \tag{6.6}$$

as discussed in Section 2.2.2. See [34-Dav70] for details about asymptotic properties. Note, our implementation of a quantile estimation method is focused on the unknown case. Equation (6.3) changes to

$$\hat{x}_{\hat{q}_j} = \frac{1}{n - l_F + 1} \sum_{i=l_F}^{n} y_{j,i} \tag{6.7}$$

Corollary 6.2.2 *$\hat{x}_{\hat{q}_j}$ is an asymptotically unbiased estimator of $F_X^{-1}(q_j)$ for large p, where $n - l + 1$ is the number of order statistics used for calculation.*

Proof $\mathrm{E}\left[y_{j,i}\right] = F_{X_i}^{-1}(q_j)$ holds for large values of p, see [33-DJ54] and [34-Dav70]. Analogously to the proof of Theorem 6.2.1, $\mathrm{E}\left[\hat{x}_{\hat{q}_j}\right] = F_X^{-1}(q_j)$ can be shown. ∎

Every simulation is a statistical experiment. Point estimators never return exact values, even if they are unbiased. Confidence intervals, or interval estimates, are essential to provide convincing results. To establish a confidence interval for x_{q_j} given by Equation (6.7) its variance $\mathrm{Var}\left[\hat{x}_{q_j}\right]$ is helpful. Note, that $\{y_{j,i}\}_{i=l_F}^{\infty}$ (row) is correlated and the variance cannot be estimated directly. The form of

the right hand side of Equation (6.3) is identical to mean value estimators of single simulation runs. The difference is that each component describes a quantile. Therefore, known techniques for variance estimation of mean value estimators can be applied. Spectral analysis and batching methods are commonly used in mean value analysis.

Both, spectral analysis and batching methods, are already used in [71-HL84] for variance estimation in quantile analysis. Heidelberger and Lewis use the maximum transformation, see Section 2.4.1, to obtain extreme quantiles of the output process. Here, we replace the maximum transformation because independent replications allow us to use the more natural estimators Equation (6.3) or Equation (6.7) Thereby we extend the method of Heidelberger and Lewis to multiple independent replications. Furthermore, the original method operates on data with fixed sample size. The extended method is applicable for a sequential approach.

6.2.1 Spectral Analysis

In [72-HW81] a confidence interval for the steady state mean value is generated by spectral analysis on basis of a single simulation run. This confidence interval is used to control run length to obtain estimates with a specified accuracy. This method assumes that the output sequence converges to a steady state behaviour which can be modelled as a covariance stationary process. It was originally used for mean value analysis. In conjunction with the maximum transformation, it is also used for estimation of one single quantile, see [71-HL84].

Similarly, the sequence $\{y_{j,l_F}, y_{j,l_F+1}, \ldots, y_{j,n}\}$ can be used for this spectral method, even though the analysed measure is a quantile and not the mean. This is because the spectral method's only assumption is that the analysed sequence of observations represents a covariance stationary process (see [72-HW81]). The spectral method of Heidelberger and Welch is applicable in this context. In the following we describe how spectral analysis can be used to establish a confi-

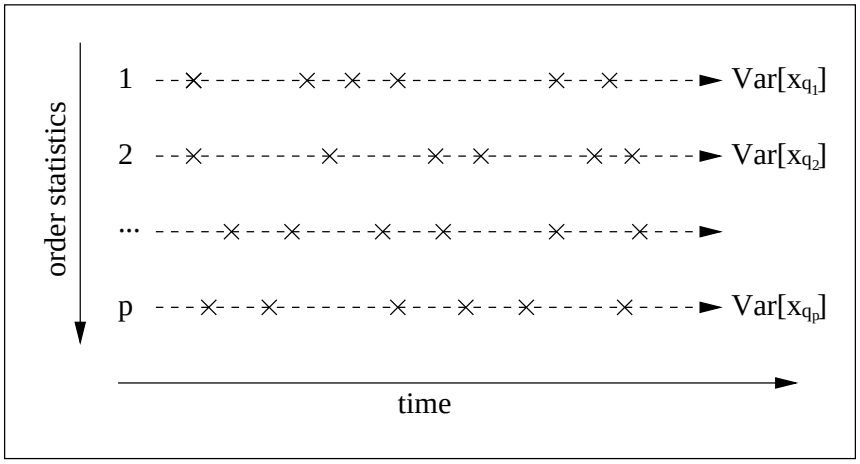


Figure 6.1: Schematic diagram of spectral analysis to estimate $\mathrm{Var}\left[\hat{x}_{q_j}\right]$, where $1 \leq j \leq p$.

dence interval for the point estimator given by Equation (6.3). Figure 6.1 shows a schematic diagram of spectral analysis applied for every sequence of order statistics.

Let $\{y_{j,i}\}_{i=l_F}^{n}$ be a realisation of the stochastic process $\{Y_{j,i}\}_{i=l_F}^{n}$. The covariance function $\gamma(k)$ is defined by

$$\gamma(k) = \mathrm{Cov}\left[Y_{j,i}, Y_{j,i+k}\right]. \tag{6.8}$$

Because the process is assumed to be covariance stationary the absolute value of i does not matter, as long as $i \geq l_F$. $\gamma(k)$ may also depend on rank j, however, to simplify the notation this dependence is dropped in the following discussion. The spectral density $\rho(f)$ at frequency f is defined as

$$\rho(f) = \sum_{k=-\infty}^{+\infty} \gamma(k) \cos(2\pi f k). \tag{6.9}$$

For $\hat{x}_q$, defined by Equation (6.3),

$$\mathrm{Var}\left[\hat{x}_q\right] = \frac{\rho(0)}{N}, \tag{6.10}$$

where $N = n - l_F + 1$, assuming that N is large. This means that a confidence interval for Equation (6.3) can be constructed if $\rho(f)$ can be estimated at $f = 0$.

This is usually done by transformation of the periodogram to a function $J(f)$, see e.g. [72-HW81] for details, and a polynomial fit. The polynomial fit is based on two parameters. The first parameter K is the number of points of $J(f)$ used to obtain the polynomial fit. The second parameter d is the degree of the polynomial. In [72-HW81] an algorithm is given to estimate $\hat{\rho}(0)$. In [93-MEP04] the standard setting $d = 2$ and $K = 25$ of this algorithm is discussed. A positive slope of $\hat{\rho}(f)$ at $f = 0$ can lead to a too small estimate of $\rho(0)$. Using the maxima of $\hat{\rho}(0)$ for $d = 0$ or $d = 2$ results in more accurate confidence intervals. Finally, a confidence interval can be derived by assuming that

$$\frac{\hat{x}(q) - F_X^{-1}(q)}{\sqrt{\frac{\hat{\rho}(0)}{N}}} \tag{6.11}$$

is governed by a t-distribution.

Using sorted values of p independent replications, p output processes $\{y_{j,i}\}_{i=l_F}^{\infty}$ are available and Equation (6.3) can be applied for all $\hat{q}_j$, see Equation (6.6), with $1 \leq j \leq p$. $\hat{x}_{\hat{q}_j}$ and $\mathrm{Var}\left[\hat{x}_{\hat{q}_j}\right]$ can be calculated for all j separately, as well as the confidence intervals based on $\mathrm{Var}\left[\hat{x}_{\hat{q}_j}\right]$.

In spectral analysis grouping observations in batches for obtaining uncorrelated batch statistics is not needed. However, we apply batch means for the purpose of reducing data. This is possible as $\mathrm{Var}\left[\hat{x}_{\hat{q}_j}\right]$ can also be estimated from the spectral density function of batch means, using batches of arbitrary size. Batch mean is explained in detail in the next section. Here, we apply batch means to reduce data, thus, no statistical test for independence of batch means is necessary. Equation (6.10) can be extended to

$$\mathrm{Var}\left[\hat{x}_q\right] = \frac{\rho(0)}{N} = \frac{\rho_B(0)}{M}, \tag{6.12}$$

if the number of batches M is large (see e.g. [72-HW81]). $\rho_B(0)$ is the power spectrum of the sequence of the batch means evaluated at zero. Based on this result we can introduce batch means into spectral analysis. Batching guarantees

that a constant storage requirement is sufficient for sequential analysis, this will also be discussed in the next section. More details on sequential spectral analysis is given in [99-Paw90], where pseudocode of the algorithms and flow charts are provided.

6.2.2 Non-Overlapping Batch Means

In the last section we described an approach using spectral analysis to estimate $\mathrm{Var}\left[\hat{x}_{\hat{q}_j}\right]$, which can be extended by batching data to obtain a sequential version. Here, we would like to describe an alternative approach using batch means, exclusively. The literature about batching methods is vast. Possibly one of the earliest described batching methods for simulation output analysis is [50-Fis78]. The basic idea is to divide the output process into subsequences of equal size, called batches. For all batches a batch statistic is calculated, e.g. the batch mean. The value of this approach is that the batch statistics become approximately independent of each other for a large batch size. The assumed near-independence helps to estimate the variance of the batch statistics. The difficulty of this method is the determination of an appropriate batch size m. The purpose of batching is either to produce nearly uncorrelated (secondary) output data or to reduce the size of output data. Here, it is important to obtain nearly uncorrelated data to estimate $\mathrm{Var}\left[\hat{x}_{\hat{q}_j}\right]$ exclusively based on batch mean. Additionally, such batching allows the reduction of the output data. This is useful in sequential analysis for achieving constant memory requirements.

For simplicity, we reduce autocorrelation of data by applying non-overlapping batch means (NOBM), since it will be shown that such an approach can produce statistically accurate estimates of quantiles. Figure 6.2 shows a schematic diagram of NOBM applied for every order statistic. The transformed data is given by

$$z_{j,i}(m) = \frac{1}{m}\sum_{k=1}^{m} y_{j,(l_F+im-k)} \tag{6.13}$$

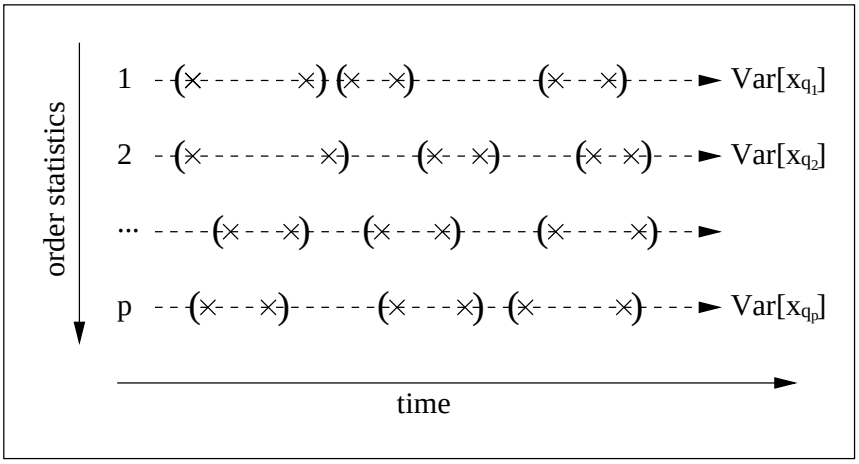


Figure 6.2: Schematic diagram of NOBM to estimate $\mathrm{Var}\left[\hat{x}_{q_j}\right]$, where $1 \leq j \leq p$.

with $1 \leq j \leq p$, $1 \leq i \leq n_b$ and $n_b = \frac{n-l_F+1}{m}$. The size of the resulting data matrix is reduced m times to pn_b. The point estimate $\hat{x}_{\hat{q}_j}$ can now be calculated by

$$\hat{x}_{\hat{q}_j} = \frac{1}{n - l_F + 1} \sum_{i=l_F}^{n} y_{j,i} = \frac{1}{n_b} \sum_{i=1}^{n_b} z_{j,i}(m). \tag{6.14}$$

This equation is a variation of Equation (6.3), because the sum is over the batch means. With an appropriate choice of m the batch means $z_{j,1}(m)$, $z_{j,2}(m)$, ... are approximately independent of each other. Under this assumption $\mathrm{Var}\left[\hat{x}_{\hat{q}_j}\right]$ can be estimated, by

$$\sigma^2_{\hat{x}_{\hat{q}_j}} = \frac{1}{n_b(n_b - 1)} \sum_{i=1}^{n_b} \left(z_{j,i}(m) - \hat{x}_{\hat{q}_j}\right)^2 \tag{6.15}$$

as in [50-Fis78]. $(\hat{x}_{\hat{q}_j} - F_X^{-1}(\hat{q}_j))/\sigma_{\hat{x}_{\hat{q}_j}}$ is approximately t-distributed with n_b degrees of freedom, thus a confidence interval can be constructed. For every $1 \leq j \leq p$ the expected value $\mathrm{E}\left[\hat{x}_{\hat{q}_j}\right]$, its variance $\mathrm{Var}\left[\hat{x}_{\hat{q}_j}\right]$ and the corresponding confidence interval can be estimated. We obtain p interval estimates of quantiles of the steady state distribution $F_X(x)$.

To estimate confidence intervals on basis of $\mathrm{Var}\left[\hat{x}_{\hat{q}_j}\right]$ for all j, representing rows of the data matrix, an overall batching approach can be performed, which operates on $\{y_{j,i}\}_{i=l_F}^{\infty}$ for all j in parallel. The determination of an appropriate

overall batch size m, which is valid for all rows $1 \leq j \leq p$, is the difficulty. In previous methods tests for independence based on runs, see [123-Sie56], or lag-1 autocorrelation, see [84-LC79], are used to detect a valid batch size. Most lag-1 autocorrelation tests assume normality, which is only approximately true. For small sample sizes complex corrections of the test statistic are done, see e.g. [52-FY97]. Therefore, we introduce a heuristic test in Appendix A.1, which is based on weaker assumptions and promises good performance in our context of determining the overall batch size m. Our purpose is to find an overall batch size m that is valid for all rows $\{y_{j,i}\}_{i=l_F}^{n}$, where $1 \leq j \leq p$. An approach to detect an appropriate value of m is described in Appendix A.1. We apply this approach by using the batch mean, as defined in Equation (6.13), as batch statistic $s_{j,i}(m) = z_{j,i}(m)$. Hereby, we do not apply this method on the original output data of the replications but on the already ordered samples, so that one row $\{y_{j,i}\}_{i=l_F}^{n}$ represents the jth order statistic. This is necessary because the batch means $z_{j,i}(m)$ of these order statistics need to be nearly independent of each other. In practise this approach appears to be easy to apply and robust for any kind of output data.

NOBM is a well known approach in the area of mean value estimation. Therefore, a sequential version of NOBM is also well known, see [99-Paw90]. For a sequential approach it is quite important to be able to include further output data into analysis without the need of increasing the storage requirement. This guarantees constant storage requirements and the sequential approach can be executed for arbitrary long simulation runs. A constant storage requirement is sufficient for sequential batch mean approaches because the batch mean with batch size $2m$ can

be calculated by grouping batch means with batch size m:

$$\begin{aligned}\frac{1}{2}\left(z_{j,i}(m)+z_{j,i+1}(m)\right) &= \frac{1}{2m}\sum_{k=1}^{m}\left(y_{j,(l_F+im-k)}+y_{j,(l_F+(i+1)m-k)}\right)\\ &= \frac{1}{2m}\sum_{k=1}^{2m}y_{j,(l_F+i2m-k)}\\ &= z_{j,i}(2m) \qquad (6.16)\end{aligned}$$

(compare with Equation (6.13)). In Section 5.4.3 we used a similar approach to obtain a constant memory requirement for a truncation point detection approach. The drawback of using the batch size $m = 2^s$, where $s = 0, 1, 2, \ldots$, is that the distance between checkpoints in detection of the batch size m grows over time. On the other hand we can be assured that the variance of the batch means is not tested unnecessarily often. Note, that a minimum number of batches n_b should be regarded. In [52-FY97] $n_b \geq 32$ is recommended and $n_b = 8$ still appears tolerable. Furthermore, a minimum batch size m should be regarded. In [2-Ada83] $m \geq 50$ is recommended. These restrictions are needed when applying NOBM exclusively, because the resulting sequence of batch means has to be nearly independent. These restrictions are not needed when applying batching in conjunction with spectral analysis, because here batching is just needed for data reduction. Batching in NOBM is available in sequential versions with constant storage requirement. More details on sequential analysis is given in [99-Paw90], where pseudocode of the algorithms and flowcharts are provided. The run time of our interval estimator based on NOBM is not an important factor. This is because we use a batch size $m = 2^s$, where $s \geq 6$ for NOBM, which means that the checkpoints for the stopping criterion are geometrically distributed. This implies that for a relatively long simulation run not many repeated tests of the stopping criterion have to be done.

6.2.3 Sequential Stopping Criteria

In sequential and automated simulation all necessary decisions must be done online. The analyst influences these methods only by parameterisation at the beginning of a simulation experiment. A sequential stopping rule is used to decide whether the number of observations, which are collected so far, is sufficient to estimate reliable results or not. For sequential mean value analysis the stopping criterion is usually based on the relative error, i.e. the standardised halfwidth of the confidence interval:

$$\frac{\Delta(n)}{\bar{X}(n)} = \epsilon(n) \leq \epsilon_{\text{max}}, \tag{6.17}$$

where $\Delta(n)$ is the halfwidth of the confidence interval of the point estimate $\bar{X}(n)$, providing that $\bar{X}(n)$ is greater than zero. Both values depend on n, the number of collected observations of a single simulation run. $\epsilon(n)$ is randomly converging to zero with increasing n, because the estimate $\bar{X}(n)$ is becoming more precise and in consequence $\Delta(n)$ is decreasing. The collection of observations is continued until the threshold ϵ_{max} is greater then the relative error $\epsilon(n)$. A common setting is $\epsilon_{\text{max}} \leq 0.1$.

When estimating the steady state probability distribution $F_X(x)$ on basis of several quantiles $F_X^{-1}(q_0), \ldots, F_X^{-1}(q_p)$ there is also a rule needed, which decides whether the estimates are statistically accurate or not. A straight forward approach is to adopt the stopping criterion used in mean value analysis also for quantile analysis. However, a serious problem arises: It is quite likely for one of the quantiles that $F_X^{-1}(q_j) \approx 0$. In this case the confidence interval would be standardised to zero. This leads to an infinite relative error and the stopping criterion cannot be fulfilled at all. Furthermore, we do not think that it is really desirable to standardise all estimated quantiles by different values. Some quantiles would consume a lot of simulation time until they are finally able to meet the threshold ϵ_{max} whereas other quantiles would fulfil the stopping criterion almost instantaneously.

The aim of a relative error is to obtain an error that is standardised by the range of interest. In mean value analysis this is the mean $\bar{X}(n)$ itself. If we estimate several quantiles $\hat{x}_{\hat{q}_j}$, see Equation (6.7), the range of interest is the observed part of $F_X(x)$, which is the range $\hat{x}_{\hat{q}_p} - \hat{x}_{\hat{q}_1}$, the difference of the highest and the lowest order statistic, see Section 2.2.1. The difference $\hat{x}_{\hat{q}_p} - \hat{x}_{\hat{q}_1}$ can never be zero, except for the trivial distribution of a fixed random variable. Therefore, we propose that the range of $\hat{F}_X(x)$ is a good standardisation:

$$\frac{\Delta_{\hat{q}_j}(n)}{\hat{x}_{\hat{q}_p} - \hat{x}_{\hat{q}_1}} = \epsilon_{\hat{q}_j}(n) \leq \epsilon_{\max}, \tag{6.18}$$

where $\Delta_{\hat{q}_j}(n)$ is the halfwidth of the confidence interval of the point estimate $\hat{x}_{\hat{q}_j}$. Note, all $\Delta_{\hat{q}_j}(n)$ are standardised by the same value, because $\hat{x}_{\hat{q}_p} - \hat{x}_{\hat{q}_1}$ does not depend on $\hat{q}_j$. To be consistent with previous notation we do not explicitly denote the dependence of $\hat{x}_{\hat{q}_j}$ on n. If a quantile fulfils the stopping criterion given by Equation (6.18) no further calculations for this quantile are needed because an estimate is found which has the demanded confidence level. No further investigation of this quantile is necessary, even though analysis of other quantiles might continue. Continuing analysis at this point might result in tiny confidence intervals which might lead to poor coverage.

6.2.4 Parameterisation

The critical parameter of NOBM is the batch size m, when applying it to receive nearly uncorrelated data. If m is too small the batch statistics $z_{j,i}(m)$, $z_{j,i+1}(m)$, ... cannot be regarded as approximately independent of each other. If m is too big the method is unnecessarily wasteful. For this reason we introduced an approach in Appendix A.1 that selects m automatically.

The remaining parameters for the batching approach are the number of replications p, the number of batches n_b, the confidence level $1 - \alpha$ and the threshold $\epsilon_{\max}$, as defined in Section 6.2.2. For experiments in later chapters we use standard

values $\alpha = 0.05$ and $\epsilon_{\text{max}} = 0.05$. The number of replications p should be an odd number, see Section 2.3, and it should be large enough with respect to the homogeneity test, see Figure 5.6 in Section 5.3.3. When using the Anderson-Darling test (see Section 5.3.2) $p = 99$ guarantees a sample of adequate size. Furthermore, we use $n_b = 128$ because this is an exponent of 2 and normality of the batch statistics can be assumed.

For spectral analysis the required parameters are the number of replications p, the number of batches M, the confidence level $1-\alpha$, the threshold ϵ_{max}, the degree of the polynomial d and the number of points K, as defined in Section 6.2.1. We use standard values $\alpha = 0.05$, $\epsilon_{\text{max}} = 0.05$, $d = 2$ and $K = 50$, as recommended in original publications. Again, we set $p = 99$ and $M = 128$ for reasons of comparison.

6.2.5 Implementation

In this section we will combine all of the topics which are discussed in this chapter so far and we will provide an implementation in pseudocode in Listing 6.1. This pseudocode is based on C++, however, differences are for example that the operator ":=" denotes an assignment and the operator "=" denotes equality. It will be shown in how far statistical techniques, for example the mean of order statistics (see Equation (6.7)), spectral analysis (see Section 6.2.1), NOBM (see Section 6.2.2) and the sequential stopping criteria (see Section 6.2.3), can be combined in one single algorithmic approach.

A simplified flowchart of quantile estimation by calculating the mean of order statistics is given in Figure 6.3. A more detailed algorithm can be found in Listing 6.1. The algorithm starts by automatically selecting a truncation point l by applying the method described in Chapter 5. Next, the minimum batch size m_{min} is selected automatically. If spectral analysis is applied, the setting $m_{\text{min}} = 1$ is adequate because no independence of batch statistics is assumed. If NOBM is

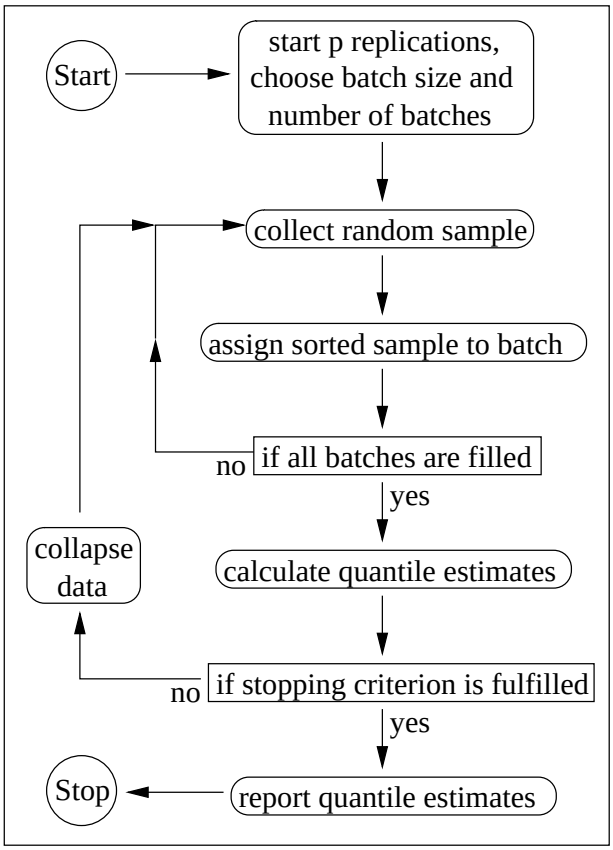


Figure 6.3: Simplified flowchart of quantile estimation by the mean of order statistics.

applied, a valid setting of $m_{\min}$ is determined by the approach described in Appendix A.1. Note that the determination of valid settings of l and $m_{\min}$ can be done in parallel to the execution of the method of Listing 6.1 to avoid restarting the simulation processes, this is not shown in Listing 6.1 for reasons of simplification. The choice of n_b and p is discussed in Section 6.2.4. In Line 5 to Line 8, variables are defined, which indicate the status of data collection, i.e. the current observation index n, batch size m, index k within a batch and index b of a batch. Their valid range is given in comment after their definition. Arrays, which are defined in Line 10 to Line 13, contain the results of the point and interval estimation of the quantiles. In Line 14, a matrix is defined which is used to store batched data.

Listing 6.1: Pseudocode for quantile estimation by the mean of order statistics.

```
int l:=autoSelect(); // truncation point
int m_min:=autoSelect(); // minimum NOBM batch size
int n_b:=128; // number of batches (must be even)
int p:=99; // number of replications

int n:=0; // 1≤n<∞ (current observation index)
int m:=1; // 1≤m<∞ (current batch size)
int k:=0; // 0≤k<m (index within batch)
int b:=0; // 0≤b<n_b (index of batch)

bool q_stop[p]:=false; // stopping criteria fulfilled
double q_prob[p]:=0; // probability of quantile
double q_pos[p]:=0; // position of quantile
double q_half[p]:=0; // halfwidth of interval estimate
double batch[p][n_b]:=0; // batched data

for (int j:=0;j<p;++j) q_prob[j]:=calculateProbability(j+1,p);

bool allStoppingCriteriaFulfilled:=false;
while (¬allStoppingCriteriaFulfilled){
   n:=n+1;
   observe(X_n);
   if (n<l) continue;

   Y_n:=sort(X_n);
   for (int j:=0;j<p;++j){ if (q_stop[j]=true) continue;
      batch[j][b]:=batch[j][b]+Y_{j,n};
   }
   k:=k+1; if (k<m) continue;

   // next batch
   k:=0; b:=b+1; if (b<n_b) continue;

   // evaluation of stopping criterion
   if (m_min≤m){
      for (int j:=0;j<p;++j){ if (q_stop[j]=true) continue;
         q_pos[j]:=calculateMean(batch[j][·]);
         double variance:=calculateVariance(batch[j][·]);
         q_half[j]:=calculateHalfwidth(variance);
      }
      double range:=q_pos[p-1]-q_pos[0];
      allStoppingCriteriaFulfilled:=true;
      for (int j:=0;j<p;++j){ if (q_stop[j]=true) continue;
         if (checkStopCriterion(q_half[j],range)) q_stop[j]:=true;
         else allStoppingCriteriaFulfilled:=false;
      }
   }
   if (¬allStoppingCriteriaFulfilled){
      collapse(batch[·][·]);
      m:=m·2; b:=n_b/2;
   }
}
```

The quantile estimation starts by calculating the position of the quantiles in the domain of the probability. The method *calculateProbability* of Line 16 implements Equation (6.6). This can be done before the collection of the data begins.

The loop in Line 19 repeats until the stopping criterion is fulfilled for all quantiles. A set of p new observations is collected in Line 21. The method *observe* applies the parallel simulation scenario of Section 3.4. Data of the transient phase is disregarded by the statement in Line 22. To receive an ordered sequence the sample of X_n is sorted in Line 24. The loop in Line 25 repeats once for each order statistic. Within this loop the current batch is updated. To explicitly indicate which value is used we added the index j to Y_n. Note, the batched data is not divided by the terms of the sum. The *if*-statement in Line 25 assures the no further calculations are done for quantiles, which already fulfil the stopping criteria. Similar *if*-statements are also used in Line 35 and Line 42. Line 28 to Line 31 update the current batch and the current index within a batch.

If all batches contain the same number of observation a checkpoint for the evaluation of the stopping criterion is reached. An evaluation is done only if the minimum batch size m_{min} is smaller than the current batch size m, see Line 34. In the loop in Line 35 the quantile estimation is done. Quantiles, which already fulfil the stopping criterion are disregarded in this step. The method *calculateMean* implements Equation (6.7). The method *calculateVariance* implements either spectral analysis (see Section 6.2.1) or NOBM (see Section 6.2.2). The method *calculateHalfwidth* calculates the halfwidth of the quantile's confidence interval. Depending on n_b the Student's t-distribution or normality is assumed. Note, the degree of freedom is also given by spectral analysis or by NOBM.

Once the estimation of quantiles is done, the stopping criterion can be tested in Line 40 to Line 45. This is done for each quantile separately. Quantiles, which already fulfil the stopping criterion are not tested again. The method *checkStopCriterion* implements Equation (6.18). If not all quantiles fulfil the stopping criterion,

more data needs to be observed. Therefore, the current batch size m is increased and b is updated. To be conform with the new batch size and to get space for more observations the matrix *batch*[·][·] is collapsed by applying Equation (6.16) in Line 48.

The run time of the method of Listing 6.1 is negligible compared to the run time which is necessary to create observations. This is because the placement of checkpoints for testing the stopping criterion is growing geometrically, i.e. the batch size m is doubled in Line 49. This assures an efficient run time if the amount of processed data is large. Storage requirements of this method are constant. The largest component is probably the matrix *batch*[·][·]. Due to the method *collapse* in Line 48 new observations can be integrated into calculations without increasing the number of batches.

6.2.6 Discussion

In this section we discuss the advantages and disadvantages of spectral analysis and NOBM when estimating $F_X(x)$. Both methods select a set of quantiles automatically. This is done on basis of the order statistics of the original output data. This is a great advantage in comparison to Raatikainen's method where the user has to select the quantiles herself or himself. In sequential analysis constant memory requirements are desirable, so that the length of simulation is not bounded by hardware restrictions, e.g. if there is not enough computer memory.

Some disadvantages remain. The estimation of the quantiles by Equation (6.7) is based on an unbounded number of samples of fixed sample size p, which is given by the number of parallel replications. As we have seen in Section 2.2.2 the estimator of the position of a quantile, Equation (2.15), Equation (2.16) and Equation (2.17), is only asymptotically unbiased. A fixed value p may lead to biased estimates for some quantiles. Furthermore, the estimated quantiles are correlated, which is not considered by the stopping criterion.

These considerations may show that the use of methods, which are well known for mean value estimation, are not the best for quantile estimation. We demonstrated how to calculate the confidence interval on basis of the variance of the sequence of order statistics by batch means and spectral analysis. Alternatively, the confidence interval could be determined by averaging confidence intervals calculated by Equation (2.13). In the next section we will demonstrate a method of quantile estimation that tries to eliminate the disadvantages of this approach.

6.3 Pooling Spaced Data

In the previous section we introduced the use of multiple replications for the estimation of quantiles. We took advantage of the independence of the replications and could solve some problems of quantile estimation of dependent data. We used estimation techniques that are well known for mean value analysis. This leads to good algorithmic properties. However, some problems remain, like the correlation between quantiles. In this section we will demonstrate how spacing of data can be used to extract random samples of almost independent and uncorrelated data from the original simulation output process. These random samples are united in a pool of data. Standard methods of quantile estimation, see Section 2.2, are applicable to this pool of data.

The first step is to find a valid truncation point, as described in Chapter 5. The remaining output process $\{X_i\}_{l_F=i}^{\infty}$ can be assumed to be in its steady state phase, where l_F is the truncation point. We can assume that all random variables of the output process $\{X_i\}_{l_F=i}^{\infty}$ are identically distributed, i.e. Equation (2.5) holds.

Approximate independence within data collected in one replication can be achieved by establishing a pool of observations, which are spaced far apart from each other. Let s be the space size. Then x_{j,l_F}, x_{j,l_F+s}, x_{j,l_F+2s}, ..., where $1 \leq j \leq p$, can be regarded as nearly independent if s is large enough for all replications. These spaced observations can be united in an overall random sam-

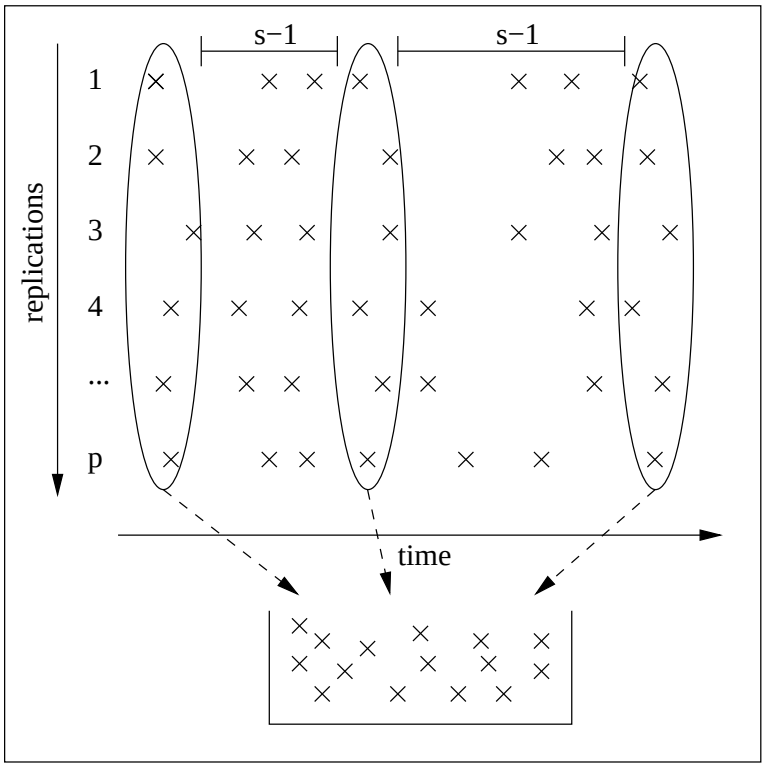


Figure 6.4: Schematic diagram of pooling spaced data.

ple by neglecting their observation index $l_F + si$ and replication index j. This leads to a pool of spaced observations, which is given by

$$\{\{x_{j,l_F+si}\}_{j=1}^{p}\}_{i=0}^{\infty}. \tag{6.19}$$

Figure 6.4 shows a schematic diagram of pooling spaced data. The size of this pool is unbounded and it contains approximately independent and identically distributed data if l_F and s are valid choices. The equations of Section 2.2.2 are now directly applicable to estimate $F_X^{-1}(q)$ because additionally Equation (2.4) holds.

The determination of an adequate value of l_F is already discussed in Chapter 5. The determination of s can be done in a similar way, as described in Appendix A.1. Spacing of output data can be seen as a special kind of batching: Here, the batch statistic is the first value of a batch, i.e. $s_{j,i}(m) = x_{j,l_F+si}$, and we define $\hat{r}^{(p)}(P_1)$ as Pearson's correlation coefficient of the original lag-1 paired batch

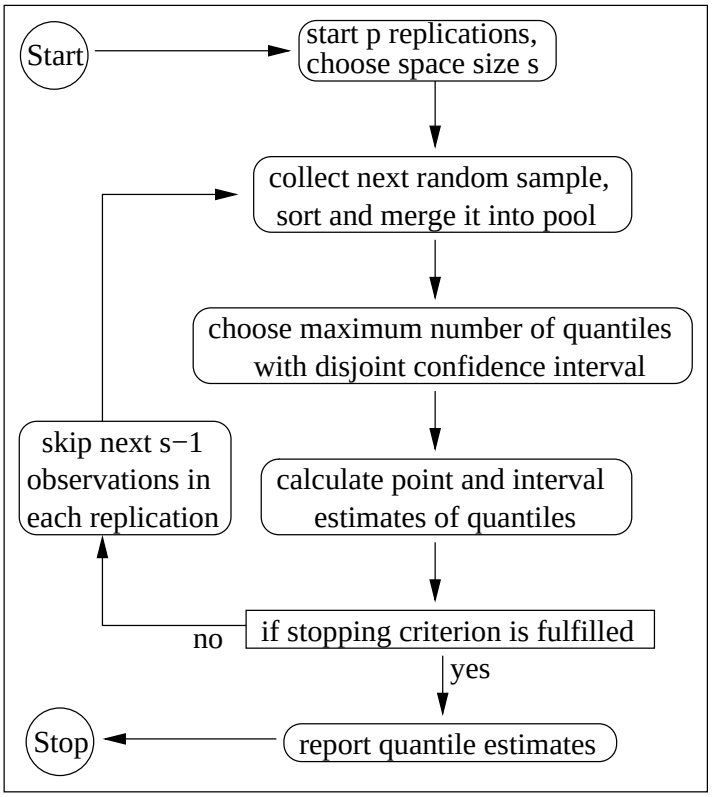


Figure 6.5: Simplified flowchart of pooling spaced data.

statistics $\left\{x_{j,l_F+si}, x_{j,l_F+s(i+1)}\right\}_{i=0}^{\infty}$. The median confidence interval is calculated on basis of permutations of the original sequence, as shown in Appendix A.1. If the assumption of independence cannot be verified the space size s can be doubled by skipping every second entry in the pool of spaced observations, followed by another test of independence. Our choice of s is the minimum value that fulfils the test of Appendix A.1 for all $1 \leq j \leq p$.

The basic idea of pooling spaced data is given by a flowchart in Figure 6.5. A more detailed description are given in Listing 6.2, which shows an implementation of pooling spaced data in pseudocode. Mainly, the pseudocode is based on the programming language C++. Note, the operator ":=" is an assignment and the operator "=" checks equality. The space size s is detected automatically by the method described in Appendix A.1. The truncation point l is detected by the

methods of Chapter 5. The number of replications p is an input parameter. The other parameters $1 \leq k \leq s$ (counter within a space), $1 \leq i \leq \infty$ (current simulation length), $1 \leq n < \infty$ (counter of spaces) and $1 \leq c < \infty$ (next checkpoint) are used to organise the methods prosecution. Note, some of these variables are initialised outside their valid range. The results of this method are given by the set of quantiles Q and by the pool of spaced data P. The use of

Listing 6.2: Pseudocode of pooling spaced data.

```
0  int s:=autoSelect(); // 1≤s<∞ (space size)
   int l:=autoSelect(); // 1≤l<∞ (truncation point)
   int p; // 30≤p<∞ (number of replications)

   int k:=0; // 1≤k≤s (index within space)
5  int i:=0; // 1≤i<∞ (current observation index)
   int n:=0; // 1≤n<∞ (index of space)
   int c:=1; // 1≤c<∞ (checkpoint)

   struct quantile{
10    double probability;
      int rank_centre;
      int rank_lower;
      int rank_upper;
   };
15
   quantile Q[]; // (set of quantiles)
   double P[]; // (pool of spaced data)

   bool StoppingCriterionIsFulfilled:=false;
20 while (¬StoppingCriterionIsFulfilled){
      i:=i+1;
      observe(X_i);
      if (i<l) continue;

25    k:=k+1;
      if (k<s) continue;
      P:=merge(P,sort(X_i));
      n:=n+1;
      k:=0;
30
      if (n<c) continue;
      Q:=calculateSetOfQuantiles(n·p);
      StoppingCriterionIsFulfilled:=checkStoppingCriterion(Q,P);
      c:=c·2;
35 }
```

efficient data structures for Q and P are quite important, because their size grows as the simulation progresses and, especially P, can be large. We recommend to use the C++ Standard Template Library and to used the template *list* to implement P because it supports sorting, merging and extending the data sample. The loop in Line 20 repeats until the stopping criterion, see Section 6.3.1, is fulfilled. A new random sample consisting of p observations is collected in Line 22. Hereby, we apply the parallel simulation scenario which is discussed in Section 3.4. Data of the transient phase is ignored, see Line 23. Line 26 checks if the space between the current observation index i and the last stored sample is large enough. If this is the case, the new collected random sample is sorted and merged into the pool of spaced data in Line 27. Note, here we store the data of the last observation index within a space (or batch). If a checkpoint is reached, see Line 31, the method of Section 2.3 is performed in Line 32 for a sample size of np observations. This method returns a set of quantiles Q with disjoint confidence intervals given by its probability and the ranks of the point estimate and the lower and upper bound of the interval estimate, see Line 9 to Line 14. These ranks point at observations in the sorted pool P. In Line 33, Q and P are examined if the estimates fulfil the stopping criterion. The worst case execution time of Listing 6.2 is of minor interest. Because of the choice of checkpoints of the stopping criterion, which are placed at observation indexes $c = 2^j$, where $j = 0, 1, 2, \ldots$, the run time of Listing 6.2 is negligible compared to the creation of observations by the simulation process itself, if efficient data structures are used. The storage requirement is of greater interest due to the growing size of Q and P. The size of Q is smaller than the size of P because Q contains only a small selection of quantiles pointing at ranks in P. More details about the size of the pool P and the run time of sorting and merging are given in the discussion about sequential aspects of this method in the following section.

6.3.1 Sequential Approach and Stopping Criteria

Quantile estimators, as for example Equation (2.15), Equation (2.16) and Equation (2.17), are usually approximately unbiased provided the sample size is large. Therefore, it is quite important for a sequential approach to be able to extend the sample size. As we have mentioned before, the sample size of $\{\{x_{j,l_F+si}\}_{j=1}^{p}\}_{i=0}^{\infty}$ is unbounded. By adding an additional sequence $\{x_{j,l_F+s(n+1)}\}_{j=1}^{p}$ of previously unprocessed observations the sample size can be extended.

For quantile estimation based on order statistics the sample has to be sorted. The most efficient way of sorting in this case is to merge two already sorted samples. Let assume that $\{\{x_{j,l_F+si}\}_{j=1}^{p}\}_{i=0}^{n}$ with size pn is already sorted. The new sample $\{x_{j,l_F+s(n+1)}\}_{j=1}^{p}$ can be sorted in $\mathrm{O}\left(p\log(p)\right)$. Merging of the two samples can be done in $\mathrm{O}\left(pn+p\right)$. So the total runtime of adding new observations to a sorted pool of spaced data is $\mathrm{O}\left(p\log(p)+p(n+1)\right)$. Because usually $n >> p$ holds, we can simplify the runtime to $\mathrm{O}\left(pn\right)$, which is efficient.

Because our aim is to estimate $F_X(x)$ on basis of several quantiles, we select quantiles as described in Section 2.3. This guarantees that the confidence intervals of the selected quantiles do not overlap and it implies that their probability mass is distributed to different order statistics, see Equation (2.13). Hereby, high correlation between quantile estimates can be avoided. How many quantiles are selected, depends on the sample size pn.

Here, a stopping criterion could be defined by simply requiring a minimum number of quantiles to be selected with disjoint confidence intervals. The bigger the sample size the more quantiles can be selected by the approach described in Section 2.3. The simulation experiment could be stopped if the sample is large enough to estimate the minimum number of quantiles with disjoint confidence intervals.

On the other hand, a stopping criterion could depend on the size of the confidence interval. Let $\Pr\left[Y_l \leq x_q \leq Y_u\right] \geq 1-\alpha$ be a balanced confidence interval

for x_q, as defined in Section 2.3. Similar to Equation (6.18), we can define

$$\frac{Y_u - Y_l}{2(Y_{pn} - Y_1)} \leq \epsilon_{\max}, \tag{6.20}$$

where Y_i denotes the ith order statistic of the pooled observations of size pn. $Y_u - Y_l$ is divided by 2 to produce the halfwidth. It is standardized by the range $Y_{pn} - Y_1$ to avoid a division by a value close to zero, as discussed in Section 6.2.3.

Pseudocode of the stopping criterion defined by Equation (6.20) is given in Listing 6.3. The variables and structures n, p, Q and P are defined as in Listing 6.2. The user specified parameter $\epsilon_{\max}$ is defined in Equation (6.20). The statement in Line 7 loops over all selected quantiles in Q. For each entry Equation (6.20) is checked in Line 10. If no quantile violates this equation the stopping criterion is fulfilled.

6.3.2 Parameterisation

Similar to the batching approach, the difficulty is estimating the space size s because it depends on the correlation of the output process. We have demonstrated an approach to determine a valid s automatically, which is similar to the determination of a valid batch size, see Appendix A.1.

Listing 6.3: Pseudocode of the stopping criterion defined by Equation (6.20).

```
0  double ε_max; // 0 < ε_max ≤ 0.01 (maximum relative error)
   int n,p;
   quantile Q[]; // (set of quantiles)
   double P[]; // (pool of spaced data)

5  double range:= P[0]-P[n·p-1];
   bool StoppingCriterionIsFulfilled:=true;
   for (int j:=0;j<sizeof(Q);++j){
      int l_rank:=Q[j].rank_lower;
      int u_rank:=Q[j].rank_upper;
10    if ((P[u_rank]-P[l_rank])/(2·range)>ε_max){
         StoppingCriterionIsFulfilled:=false;
         break;
      }
   }
```

The only remaining parameters are the confidence level $1-a$ of the confidence intervals and the threshold ϵ_{max} of the stopping criterion. We use $\alpha = 0.05$ and $\epsilon_{\text{max}} = 0.05$.

Note, this method can deal with any number p of replications. However, for the detection of the truncation point l_F a minimum of $p > 30$ is needed, see Chapter 5.

6.3.3 Discussion

The first advantage of the quantile estimation method of this section is its simplicity. Only standard methods for quantile estimation are used. Furthermore, the parameterisation is straight forward. Quantiles of $F_X(x)$ are chosen automatically so that high correlation between them is avoided. This is a great advantage compared to the methods described in Section 6.2.

The disadvantage of this method is that the memory requirement is not constant. The size pn of the pooled data does grow during simulation. However, this should not be a problem because pn does not depend on the correlation of the original simulation output process due to the space s. The size pn depends on ϵ_{max}. Experiments show that for e.g. $\epsilon_{\text{max}} = 0.05$ the needed sample size is much smaller than the usual size of computer memory.

6.4 Validation and Comparison

In Section 6.1 we discussed Raatikainen's method, which is frequently used for estimation of proportions. We noted that this method has adequate statistical properties. However, as already pointed out, it is not the best choice for automated analysis of the steady state distribution function. Therefore, we demonstrated new or extended methods for quantile estimation in Section 6.2 and Section 6.3. The properties of these methods, like e.g. low bias, are already discussed. Here, we would like to test these methods on selected examples and study their properties

experimentally.

In the next section details of our experimental analysis are given. We will use different techniques of assessing the quality of estimation results. The results of our comparative studies of these techniques are obtained by applying them to a set of representative scenarios.

6.4.1 General Approach

To be able to assess quality of the methods used for estimating quantiles we are limited to examples with known or analytically tractable steady state behaviour. For every example we plot the estimated distribution function, i.e. the estimated set of quantiles connected by a simple line, to give a visual impression of the results. This distribution is derived from the results of only one single simulation experiment. The seed of this simulation experiment is randomly selected. To have better insight into the results, we compare the known probability distribution function with the estimated probability distribution function by a Q-Q (quantile-quantile) plot, whenever it appears to be useful .

Another possibility would be to compare the estimated CDF with the known CDF by a homogeneity test. However, this approach is not advisable. The homogeneity test will never reject the hypothesis of equality in distribution of random samples, because all X_i with $i \geq l_F$ are already proven to be identically distributed among each other by this homogeneity test. This is why we do not apply a homogeneity test at this stage. We will focus on each estimated quantile instead.

A more accurate way of assessing the statistical quality of the method used for estimation of quantiles is to apply coverage analysis of their confidence intervals. This is done according to [103-PME98]. The coverage analysis is done sequentially until a certain precision of the point estimate of the coverage is reached and a minimum number of "bad" confidence intervals have been detected. A bad confidence interval is one that does not contain the theoretical result. The precision is

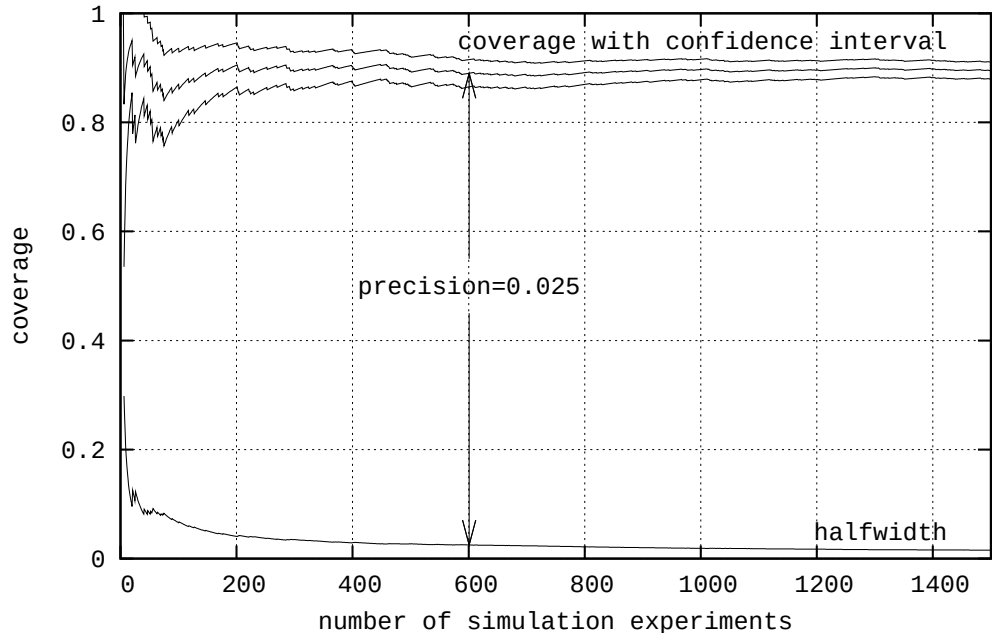


Figure 6.6: Evolution of the coverage of a quantile.

here the halfwidth of the coverage's confidence interval, which is given by

$$z_{1-\frac{\alpha}{2}}\sqrt{\frac{c(1-c)}{n_c}}, \tag{6.21}$$

where z_q is the q-quantile of the standard normal distribution, c is the coverage and n_c is the number of coverage experiments. Here, the threshold for acceptable precision is taken to be 0.025 at 0.95 confidence level. This assures that the coverage convergence curves reach a stable level for the examples in this section, as one can see in Figure 6.6. In this figure an example of the convergence of the coverage is depicted, the abscissa shows the number of simulation experiments conducted for coverage analysis and the ordinate shows the current value of coverage. Additionally, the halfwidth and the confidence interval of the coverage is plotted. The initial part of the curve shows that the convergence is not monotonic because each bad confidence interval leads to a sudden drop off in the curve. However, we can see that at a precision at 0.025 the impact of a bad confidence interval is limited

and the curve appears to be flat and we use this value for all our experiments. We did coverage analysis of this kind for all experiments in this section, the convergence of the coverage is in all cases similar to Figure 6.6 and, therefore, we will not depict it separately. In addition to coverage analysis we plot the distribution of the estimated quantiles and compared their average with the known position whenever we thought it might provide a deeper inside.

The size of the horizontal error bars in the figures showing the coverage, e.g. Figure 6.10, is calculated by Equation (6.21). We operate with a predefined precision and, thus, the error bars have the same length, in general. Exceptions occur either if it is difficult to observe bad confidence intervals or if it is difficult to observe good confidence intervals. In both cases the wanted precision might be reached very fast so that it is necessary to continue coverage analysis until a suitable amount of both, bad and good confidence intervals, is collected. An example, where it is difficult to observe bad confidence intervals, is given in Figure 6.14(c). The opposite situation, where it is difficult to observe good confidence intervals, is given in Figure 6.23(b): compare the lowest depicted quantile with error bars with all other quantiles. This explains why the length of the error bars might vary from method to method. The length of error bars might also vary for the different quantile estimates of the same method. This is because coverage analysis was performed in parallel on all selected quantiles and was stopped when each quantiles reached the predefined precision. Coverage analysis for quantiles, which reached the predefined precision earlier, was continued for reasons of simplicity.

We limit our discussion to statistical accuracy of estimates and do not investigate the efficiency in terms of the number of used observations in detail. This is because our analysis is done at checkpoints placed with geometrically growing distance, which makes a comparison difficult. The speed of collecting output data using multiple replications is also difficult to compare with the speed of data collection using one single run. Furthermore, the execution time of one simulation

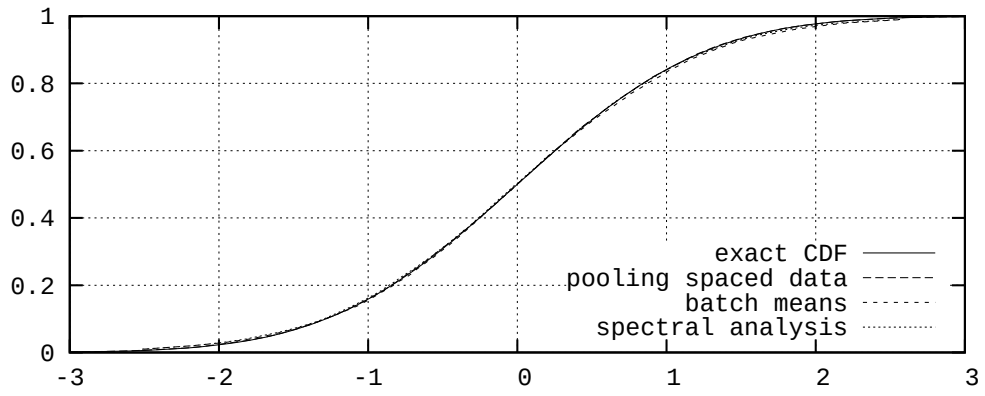


Figure 6.7: Exact and estimated CDFs of basic processes with normal distribution: $\mathrm{N}(x; 0, 1)$.

experiment of the following examples is rather short and not an issue anyway.

6.4.2 Basic Processes

In this section we test the methods of Section 6.2 and Section 6.3 on some very basic processes. These basic processes do not have a transient period and they are not autocorrelated. Therefore, they provide observations which are independent and identically distributed. The trivial choices of the truncation point $l_F = 1$, the batch size $m = 1$ and the space size $s = 1$ are adequate and selected by the method of Appendix A.1 automatically. Furthermore, the periodogram is flat, which is used in spectral analysis. This allows us to test the quantile estimators themselves, independent of other parts of the methods, on basis of independent and identically distributed data. The steady state probability distribution function of those basic processes is normal, uniform or exponential.

In our first example all X_i are taken from the standard normal distribution:

$$\forall(1 \leq i \leq \infty) : F_{X_i}(x) = \mathrm{N}(x; 0, 1)\,. \tag{6.22}$$

Here, the quantile estimators based on NOBM and spectral analysis use Equation (2.17). The quantile estimator based on pooling of spaced data always uses

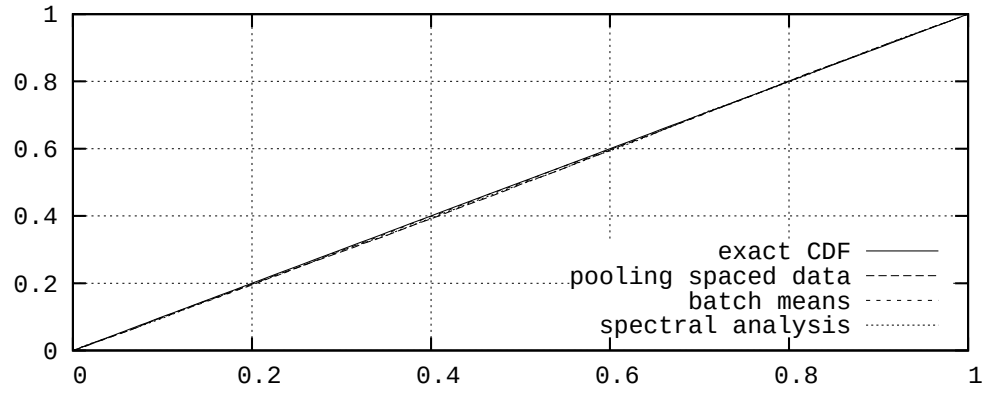


Figure 6.8: Exact and estimated CDFs of basic processes with uniform distribution: $\mathrm{U}(x; 0, 1)$.

Equation (2.15). Because its pool of observations is growing over time this is a valid approach (compare [33-DJ54]). The estimated probability distribution functions of all methods are depicted in Figure 6.7. Very little difference to $\mathrm{N}(x; 0, 1)$ can be seen. All estimated functions seem to be accurate estimates. In Figure 6.10 the coverage of all quantile estimates of each method is depicted separately. The abscissa shows the position of the quantiles in the range of probability. The ordinate shows the coverage of the belonging confidence interval. The expected coverage is 0.95. To achieve a clear arrangement we depicted the confidence interval of the coverage for selected quantiles only. We can see that the coverage of nearly all estimates is as expected. Exceptions are the estimates of extreme quantiles of the NOBM approach and spectral analysis. Those methods depend on a fixed sample size and even though they use the more specialised Equation (2.17) the extreme quantiles are biased. Extreme quantiles are those who are located in areas where the probability density function $f_X(x)$ is a value close to zero. The normal distribution is tailed towards $-\infty$ and ∞. So, extreme quantiles are located on both sides of the distribution.

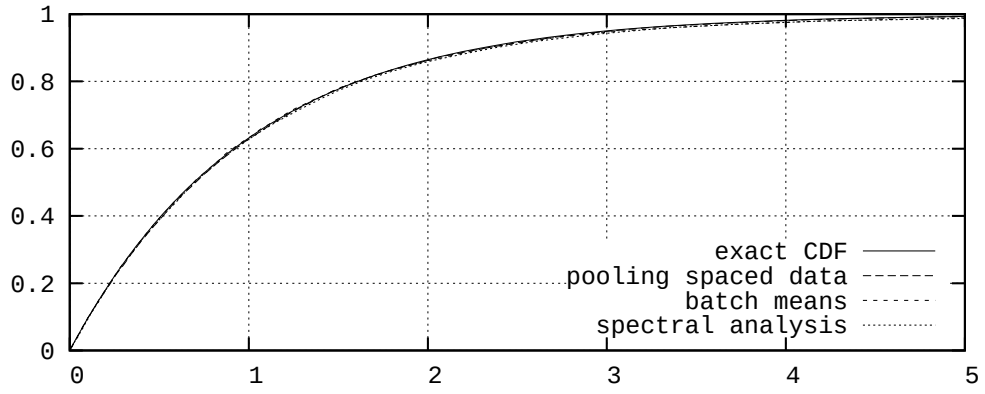


Figure 6.9: Exact and estimated CDFs of basic processes with negative exponential distribution: $\mathrm{Exp}\,(x;1)$.

All X_i are taken from the uniform distribution in our second example:

$$\forall(1 \leq i \leq \infty) : F_{X_i}(x) = \mathrm{U}\,(x;0,1)\,. \tag{6.23}$$

In this example all methods use Equation (2.15). The estimated probability distribution functions appear to be indistinguishable from $\mathrm{U}\,(x;0,1)$, see Figure 6.8. And the results of the coverage analysis is even better, see Figure 6.11. All results are as expected, as in this case there are no extreme quantiles.

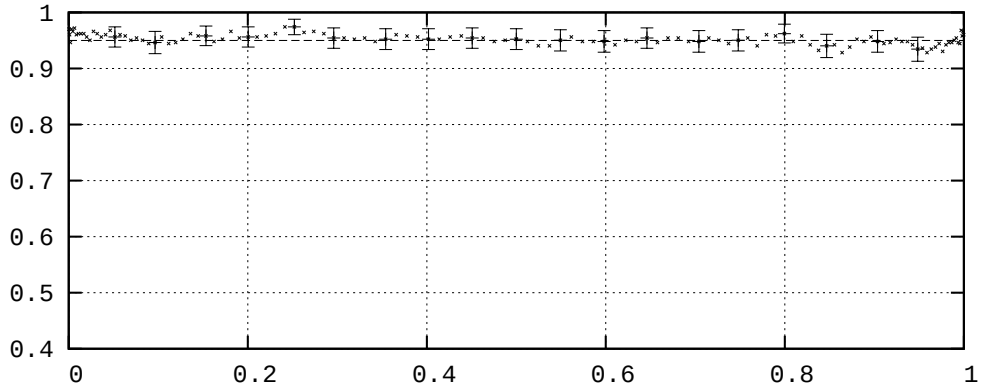
Our third example is done with all X_i negative exponentially distributed:

$$\forall(1 \leq i \leq \infty) : F_{X_i}(x) = \mathrm{Exp}\,(x;1)\,. \tag{6.24}$$

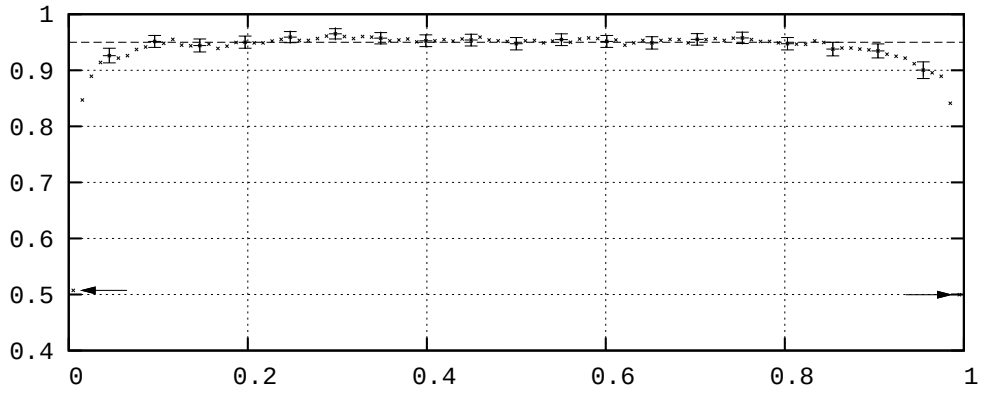
The NOBM approach and spectral analysis use Equation (2.16), the pooling approach uses Equation (2.15). The estimated probability distribution functions are depicted in Figure 6.9. Again, there seems to be no discernible difference to $\mathrm{Exp}\,(x;1)$. Figure 6.12 shows that the coverage of nearly all quantile estimates is as expected. Again, the NOBM approach and spectral analysis have problems with extreme quantiles, which are located on the right side of the distribution.

We can say that all estimated probability distribution functions are close to the exact distribution, even though the approach of Section 6.2 shows some prob-

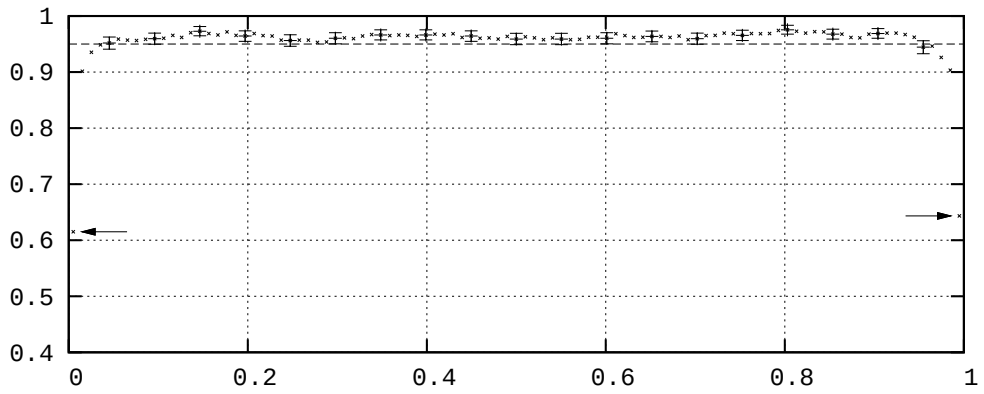
lems with extreme quantiles. The pooling approach estimates all quantiles with expected coverage and is therefore the most advisable method for these basic processes. These quantile estimation methods are designed for autocorrelated output processes, however, these examples show that they are applicable on uncorrelated output processes.



(a) pooling of spaced data

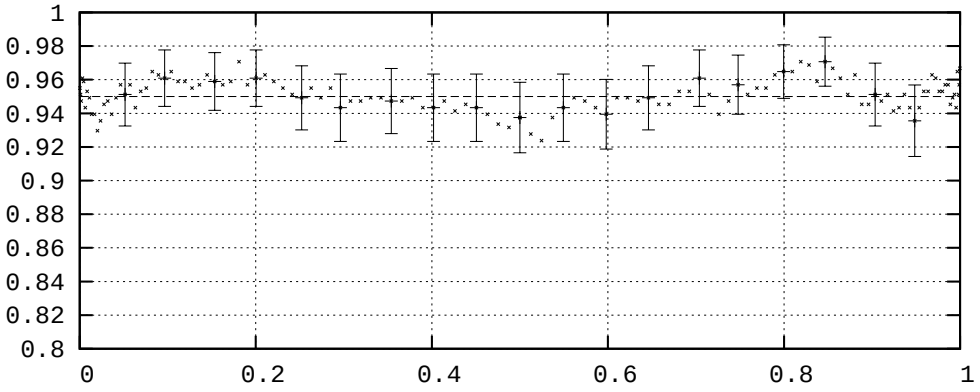


(b) NOBM

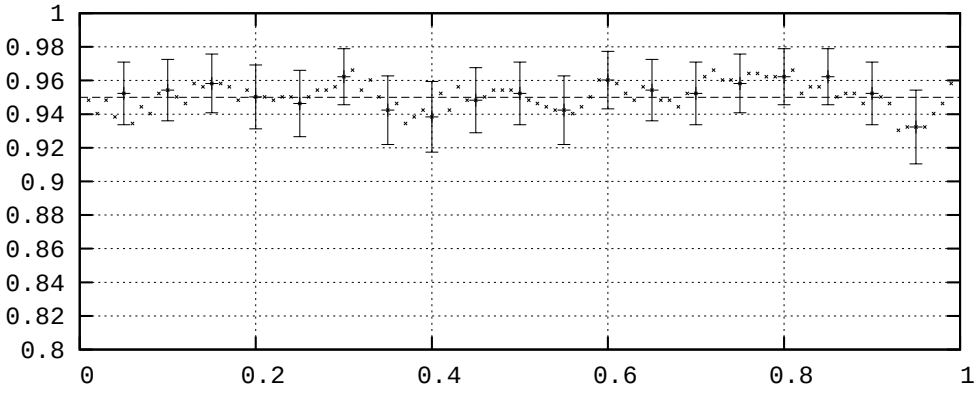


(c) spectral analysis

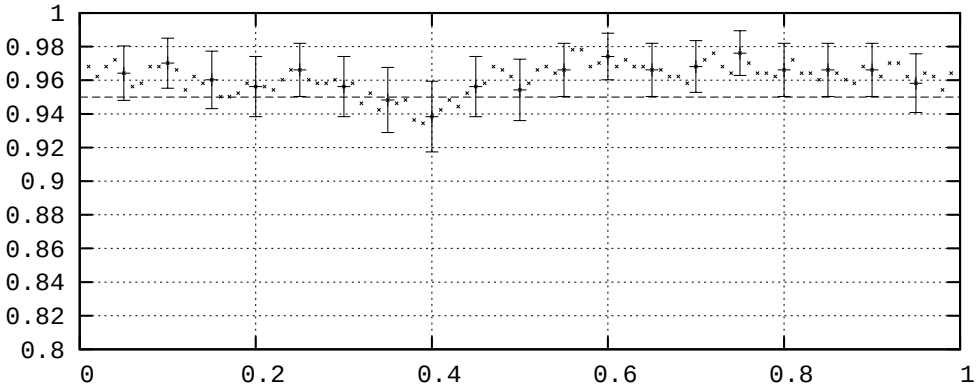
Figure 6.10: Coverage (ordinate) of the q-quantile (abscissa) of a basic process with normal distribution $\mathrm{N}\,(x; 0, 1)$.



(a) pooling of spaced data

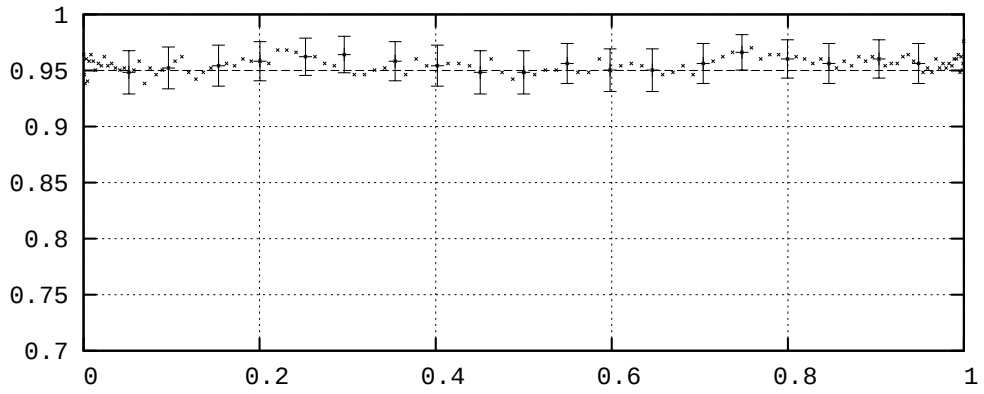


(b) NOBM

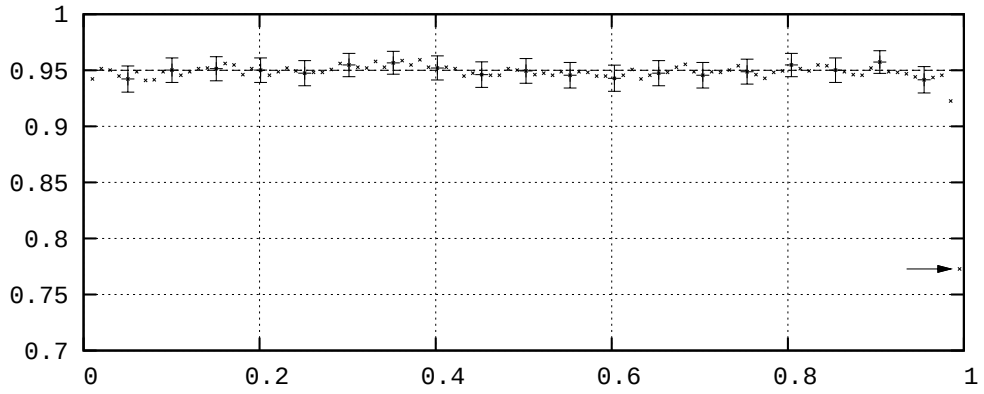


(c) spectral analysis

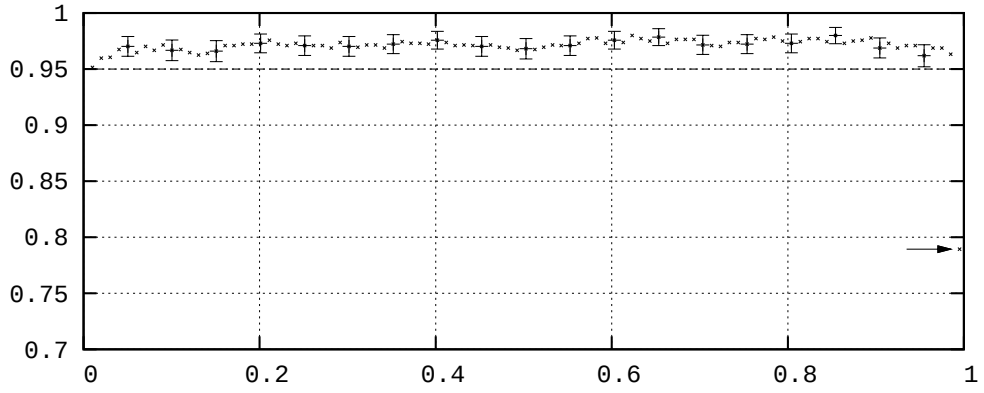
Figure 6.11: Coverage (ordinate) of the q-quantile (abscissa) of a basic process with uniform distribution $\mathrm{U}(x; 0, 1)$.



(a) pooling of spaced data

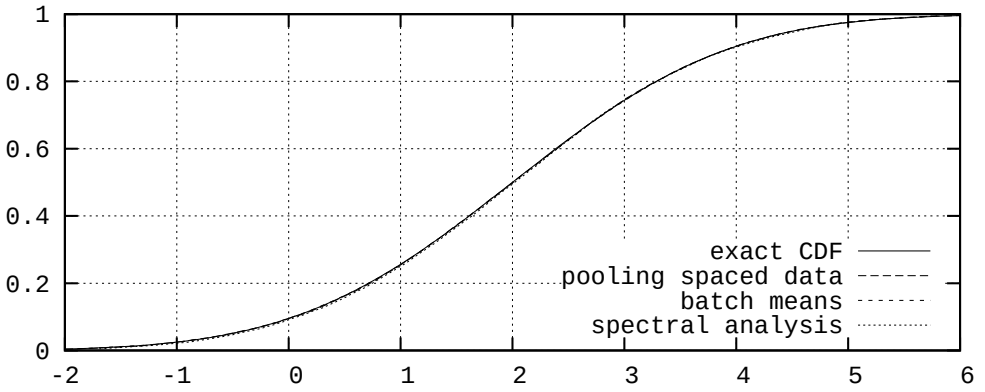


(b) NOBM

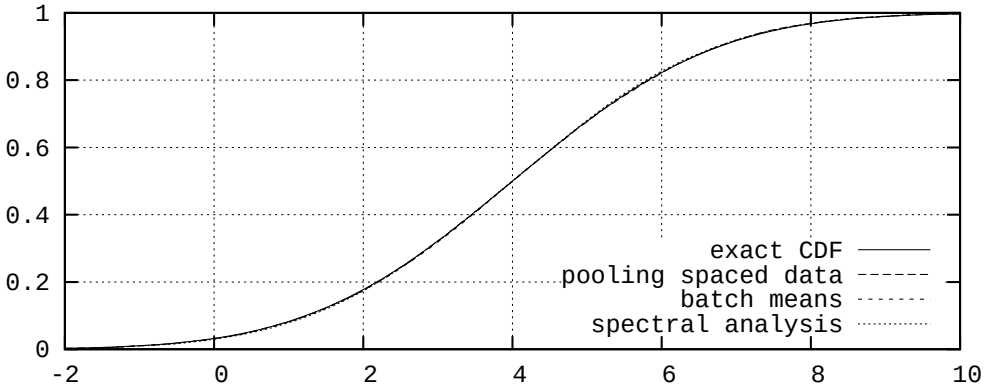


(c) spectral analysis

Figure 6.12: Coverage (ordinate) of the q-quantile (abscissa) of a basic process with exponential distribution $\mathrm{Exp}\,(x; 1)$.



(a) order $k = 1$



(b) order $k = 2$

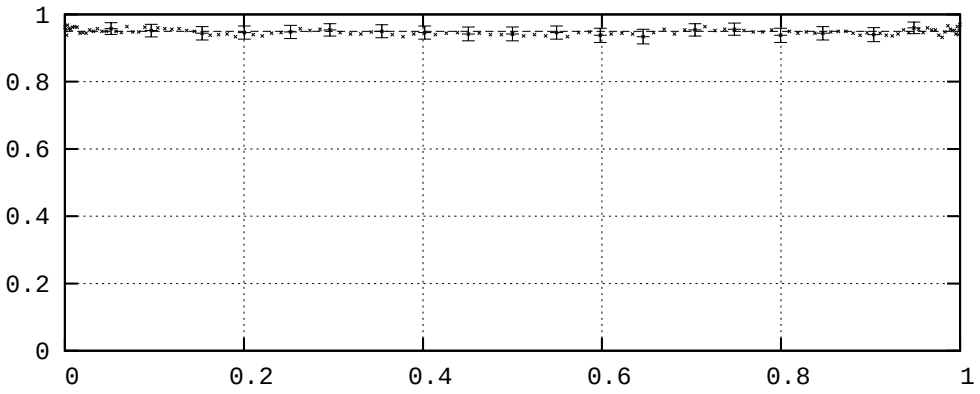
Figure 6.13: Exact and estimated CDFs of a geometrical ARMA(k, k) process.

6.4.3 ARMA Processes and M/M/1 Queues

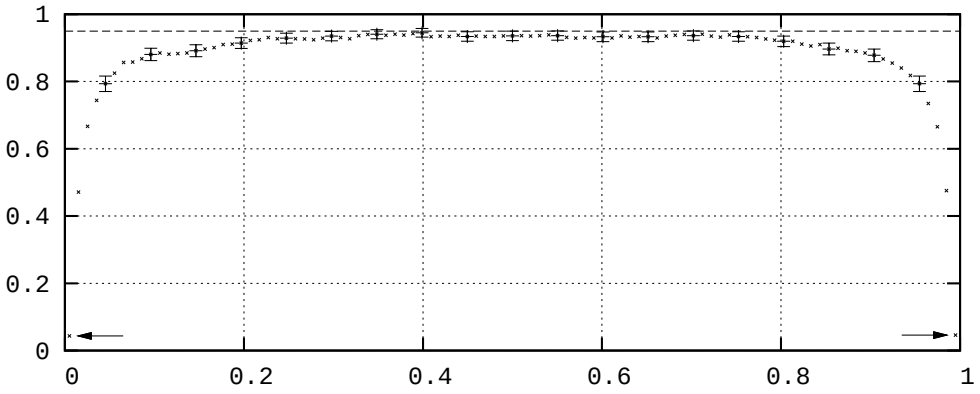
Let us look at cases where special estimators for the underlying distribution functions are known. In this section we perform quantile estimation on processes similar to those that arise in simulation models. Their output processes are autocorrelated and show an initial transient behaviour. Their steady state distribution functions follow Equation (2.17) or Equation (2.16). In contrast to the experiments of the previous section, here, methods are needed to receive secondary data, which is independent and identically distributed.

As first example we use the geometrical ARMA(k, k) process, which is de-

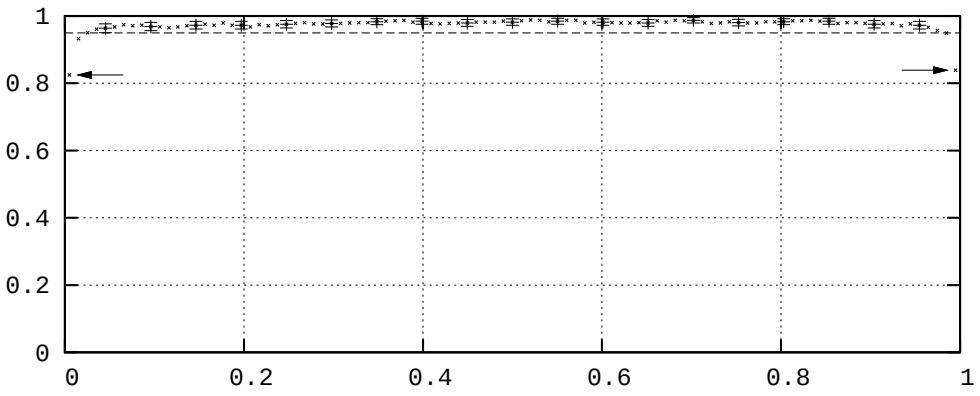
fined in Equation (A.11). We prove in Appendix A.2 that for order $k = 1$ the steady state distribution function is $\mathrm{N}\left(x; 2, \frac{7}{3}\right)$ and for order $k = 2$ it is $\mathrm{N}\left(x; 4, \frac{117}{25}\right)$. The NOBM approach and spectral analysis use Equation (2.17), the pooling approach uses Equation (2.15). In Figure 6.13 the estimated CDFs are compared with the exact CDF. All estimated distributions appear to be very close to the exact distribution. Even though the geometrical ARMA(k, k) process is strongly autocorrelated for order $k = 2$, no difference in quality of the estimates can be seen. In Figure 6.14 and Figure 6.15 the coverage of the quantile estimates is depicted for all methods separately. We can see that the coverage of extreme quantiles, which are estimated by the NOBM approach, is not as expected. The coverage of all quantiles which are estimated by the pooling approach is as expected, even the coverage of extreme quantiles. A typical space size when pooling spaced data is $s = 4$ (for $k = 1$) and $s = 16$ (for $k = 2$). Typical values of the minimum batch size when using NOBM are $m = 16$ (for $k = 1$) and $m = 32$ (for $k = 2$). When performing spectral analysis no batching was necessary. Because the batch and the space size are growing geometrically we state typical values only.



(a) pooling of spaced data

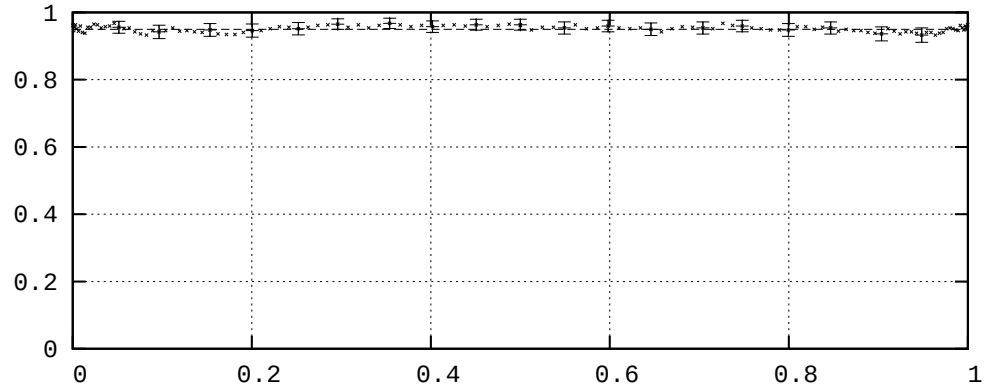


(b) NOBM

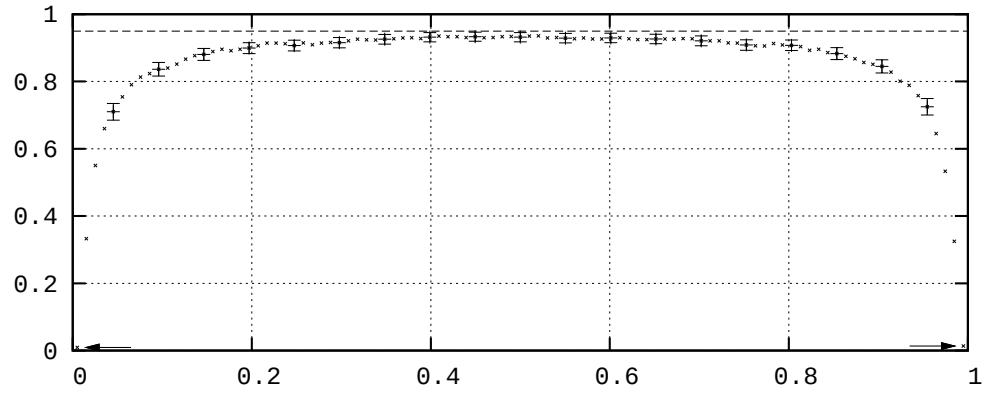


(c) spectral analysis

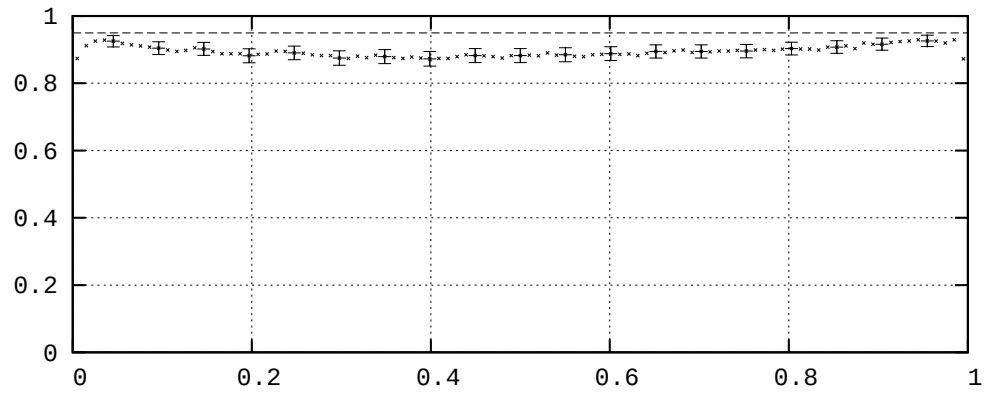
Figure 6.14: Coverage (ordinate) of the q-quantile (abscissa) of a geometrical ARMA$(1,1)$ process with normal distribution.



(a) pooling of spaced data



(b) NOBM



(c) spectral analysis

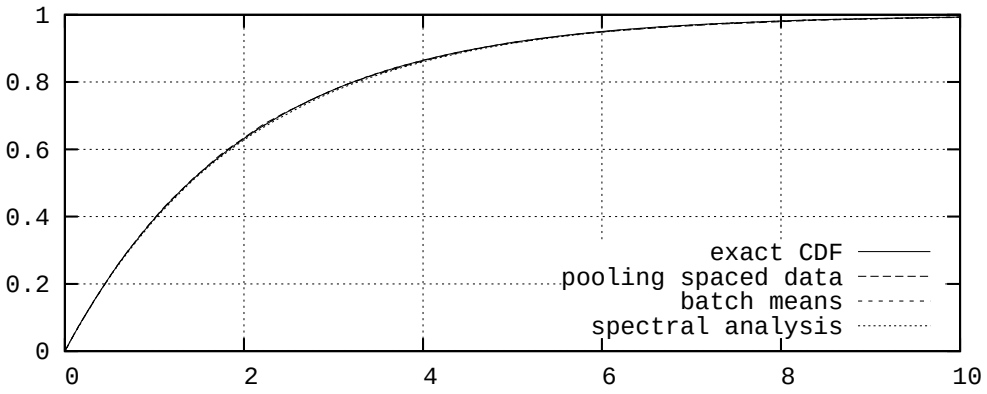
Figure 6.15: Coverage (ordinate) of the q-quantile (abscissa) of a geometrical ARMA$(2, 2)$ process with normal distribution.

In our next set of experiments we use the response time of an M/M/1 queue as the output process X_i. The simulation starts with an empty queue and idle server. We simulated this queueing system assuming different traffic intensities $\rho = \frac{\lambda}{\mu}$, where $\lambda = \{0.5, 0.75, 0.9\}$ and $\mu = 1$, to obtain output processes with low, medium and high autocorrelation. Higher values of ρ are difficult to access because they lead to extremely long simulation runs. The estimated steady state distribution functions are depicted in Figure 6.16. Comparing these estimates with the known steady state distribution function $F_{R_\infty}(x) = 1 - e^{-x\mu(1-\rho)}$, see [75-Jai91], we can detect nearly no difference. The NOBM approach and spectral analysis are using the more specialised Equation (2.16), whereas the pooling approach simply uses Equation (2.15). Even the estimates at high traffic intensity $\rho = 0.9$ are very close to the exact distribution. To get a deeper insight we performed coverage analysis of the confidence intervals of the estimated quantiles (see Figure 6.17, Figure 6.18 and Figure 6.19). Here, we can see again that the coverage is as expected, in general. However, the coverage of extreme quantiles estimated by the NOBM approach and spectral analysis is lower as expected. The coverage of quantiles which are estimated by the pooling approach is always as expected. A typical space size when pooling spaced data is $s = 16$ (for $\rho = 0.5$), $s = 64$ (for $\rho = 0.75$) and $s = 512$ (for $\rho = 0.9$). Typical values of the minimum batch size when using NOBM are $m = 16$ (for $\rho = 0.5$), $m = 256$ (for $\rho = 0.75$) and $m = 1024$ (for $\rho = 0.9$). When performing spectral analysis typically batches of size $m = 4$ (for $\rho = 0.5$), $m = 32$ (for $\rho = 0.75$) and $m = 64$ (for $\rho = 0.9$) were used just to reduce storage requirements, they are not essential to spectral analysis.

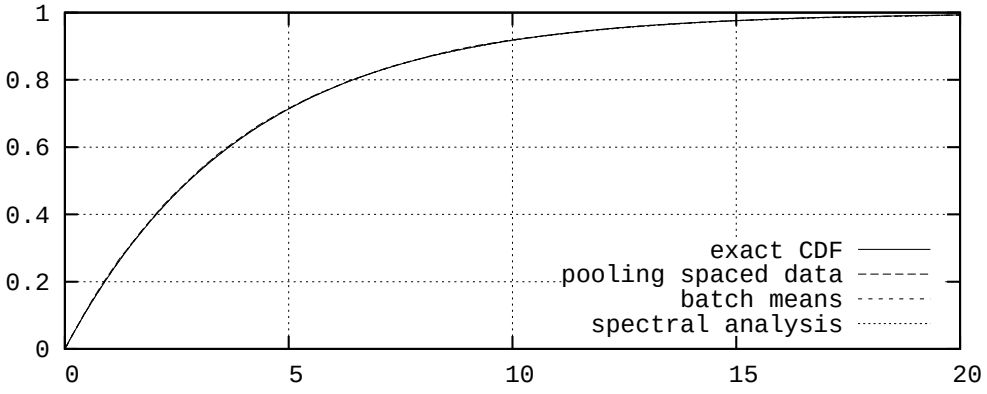
To show the dependence of the coverage on the strength of autocorrelation we varied the traffic intensity ρ and we performed more simulation experiments with $\lambda = \{0.5, 0.55, 0.6, \ldots, 0.95\}$ and $\mu = 1$. We calculated the coverage of the estimated median and depicted these results in terms of the traffic intensity ρ in

Figure 6.20. The pooling approach shows good coverage, almost independent of ρ. The coverage of the NOBM approach is a little bit lower than expected and shows a decreasing trend towards higher ρ. The coverage of spectral analysis is for low and medium traffic intensities even a little bit higher than expected. This might be caused by batching of the data during spectral analysis. For higher traffic intensities it decreases faster than the curve of the NOBM approach. Again, the pooling approach seems to deliver the best results.

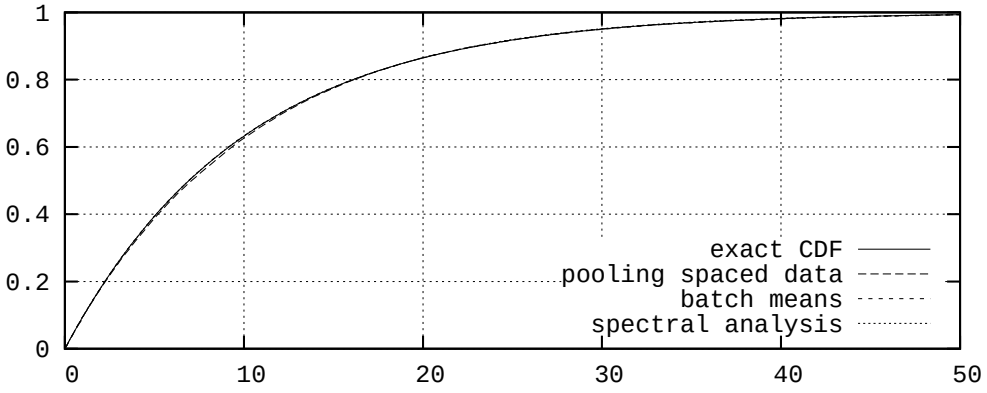
The experiments in this section show that all three methods are able to deal with transient and autocorrelated output processes. The estimated probability distribution functions are very close to their expectation. Coverage analysis reveals that coverage of extreme quantiles estimated by NOBM and spectral analysis are not as expected, these estimates can be biased. The pooling approach delivers the best results, even though both other methods are more specialised to the given distributions by using Equation (2.16) and Equation (2.17).



(a) traffic intensity $\rho = 0.5$

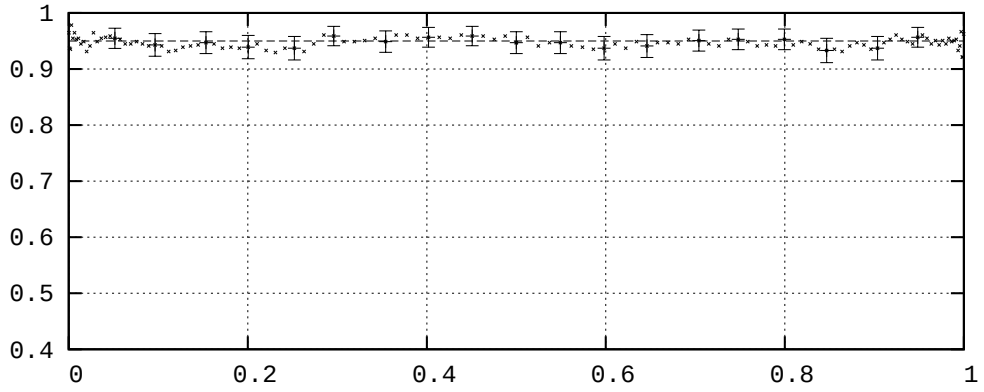


(b) traffic intensity $\rho = 0.75$

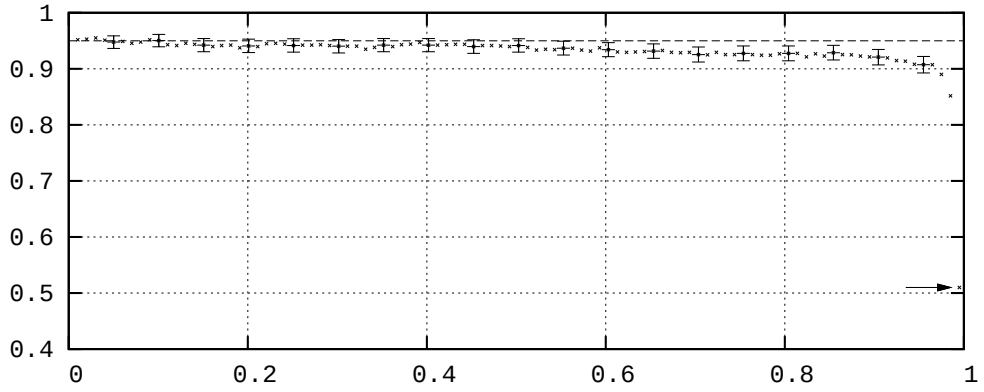


(c) traffic intensity $\rho = 0.9$

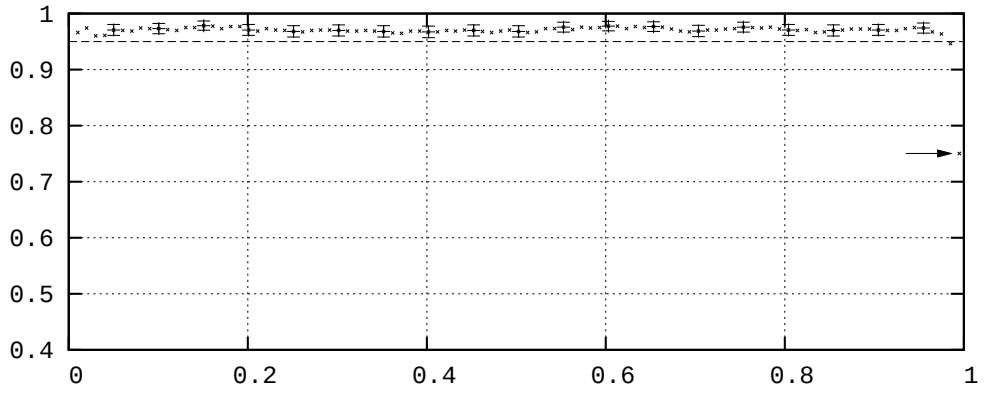
Figure 6.16: Exact and estimated CDFs of the response time of an M/M/1 queue with various traffic intensities ρ.



(a) pooling of spaced data

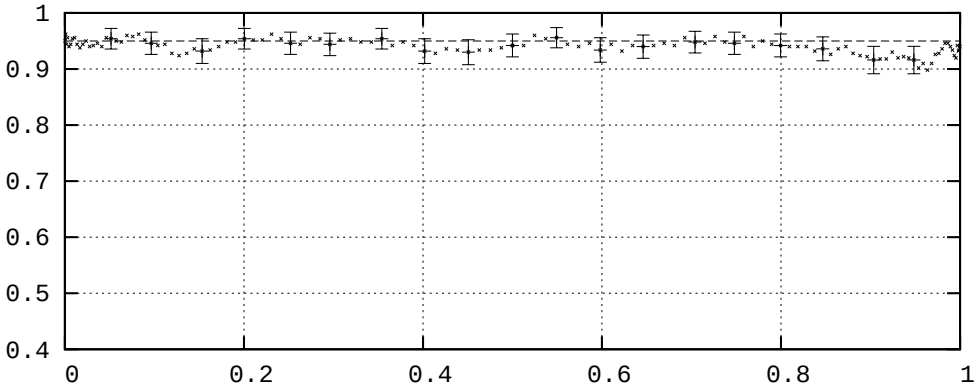


(b) NOBM

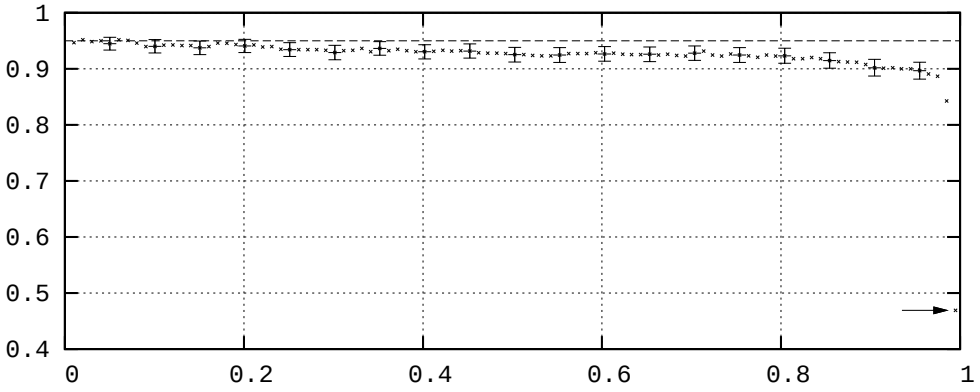


(c) spectral analysis

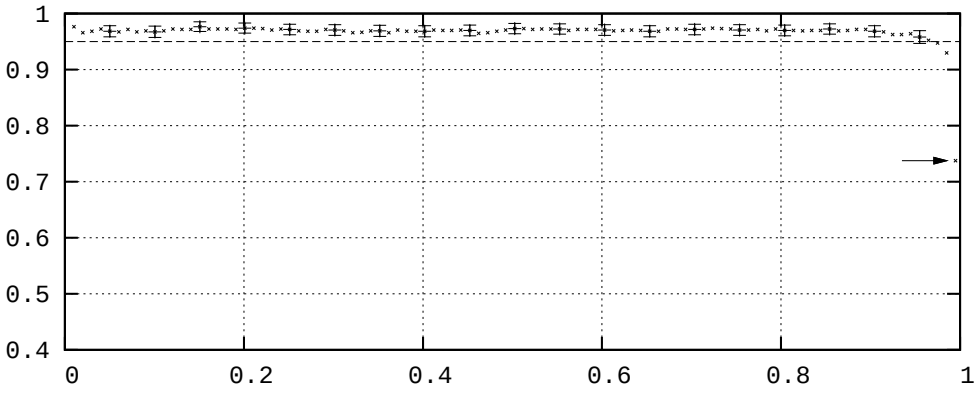
Figure 6.17: Coverage (ordinate) of the q-quantile (abscissa) of the response time of the M/M/1 queue with traffic intensity $\rho = 0.5$.



(a) pooling of spaced data

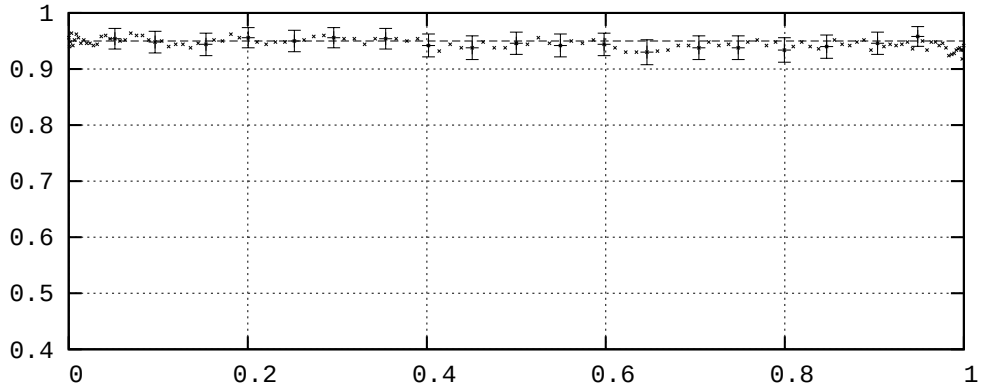


(b) NOBM

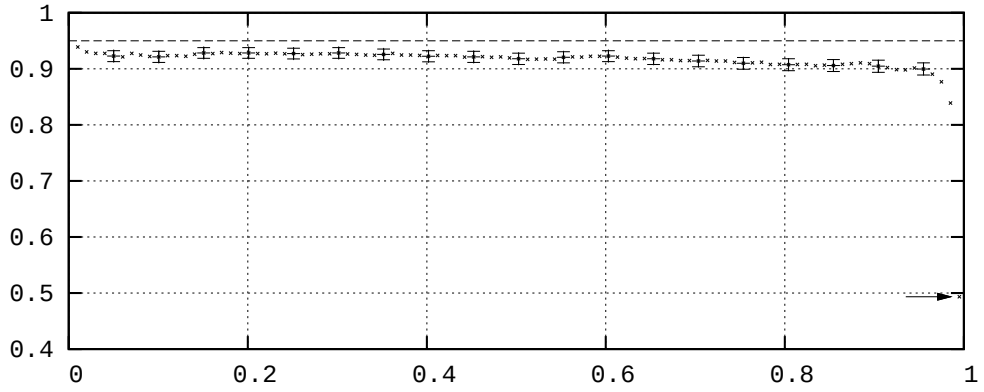


(c) spectral analysis

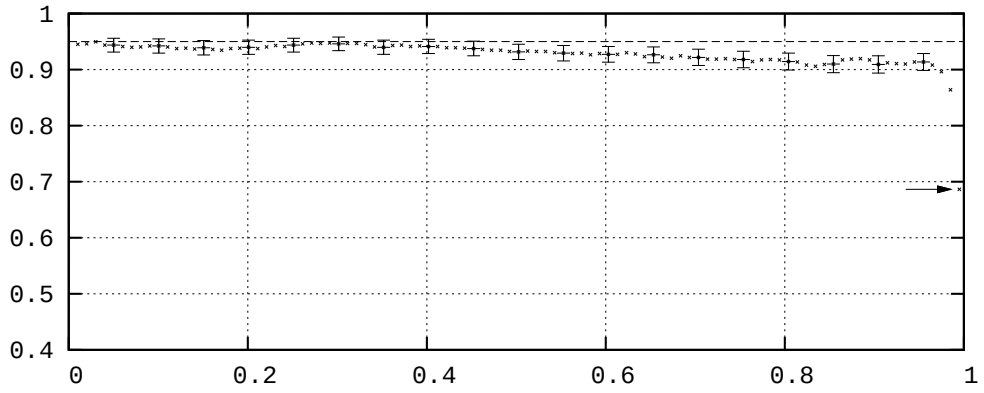
Figure 6.18: Coverage (ordinate) of the q-quantile (abscissa) of the response time of the M/M/1 queue with traffic intensity $\rho = 0.75$.



(a) pooling of spaced data

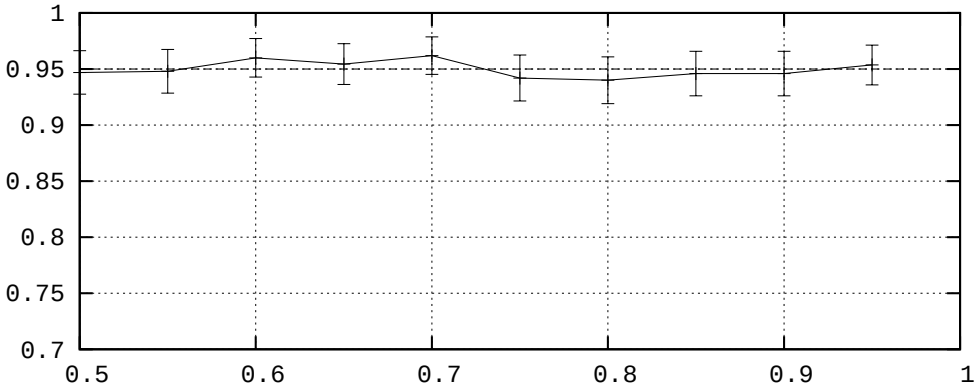


(b) NOBM

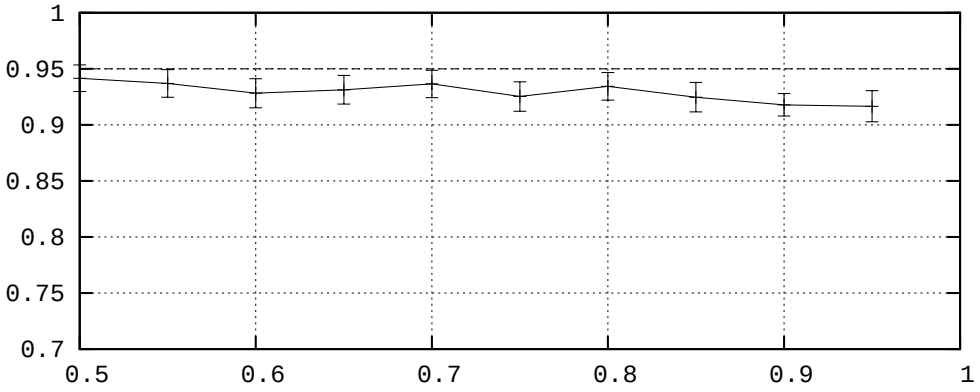


(c) spectral analysis

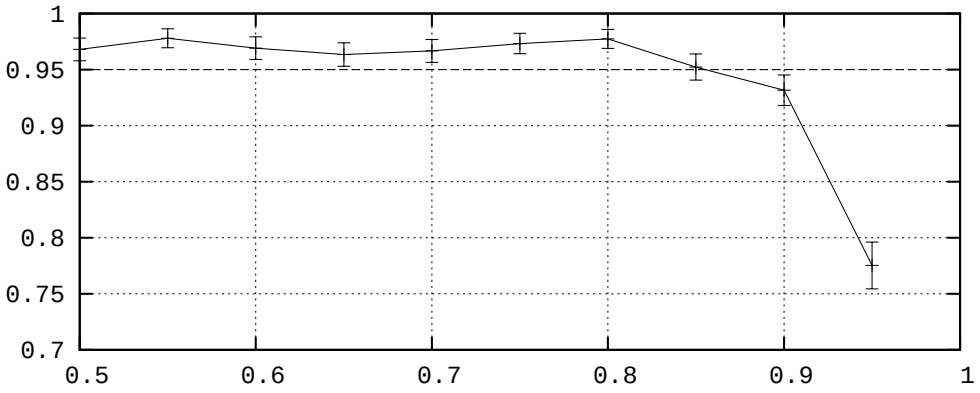
Figure 6.19: Coverage (ordinate) of the q-quantile (abscissa) of the response time of the M/M/1 queue with traffic intensity $\rho = 0.9$.



(a) pooling of spaced data



(b) NOBM



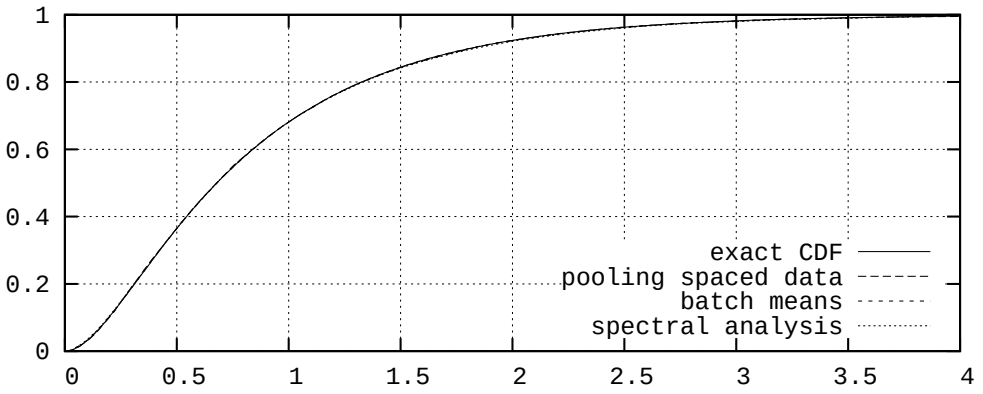
(c) spectral analysis

Figure 6.20: Coverage in dependence of the traffic intensity ρ of the median of the response time of the M/M/1 queue.

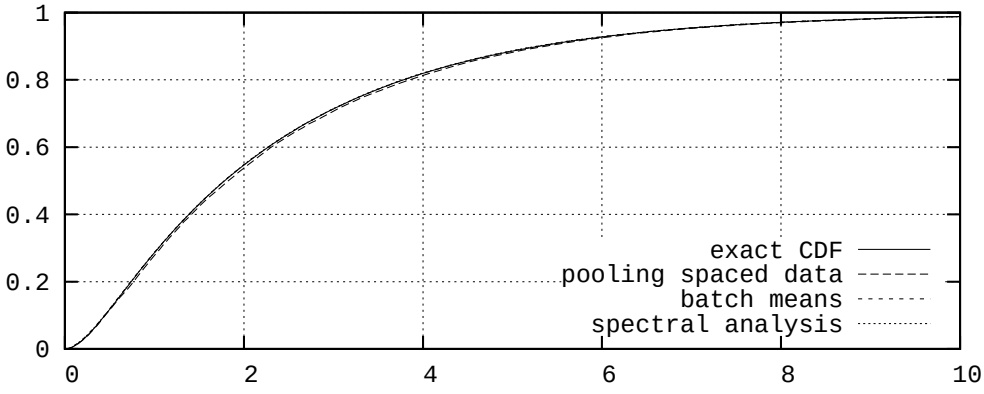
6.4.4 M/E_2/1 and M/H_2/1 Queues

Let us look at examples where no special estimators for the underlying distributions are known. In this section we will perform experiments with M/E_2/1 and M/H_2/1 queues. Their output processes are transient and autocorrelated. Furthermore, their steady state distribution function does not fit optimally to Equation (2.15), Equation (2.16) or Equation (2.17), because it is not a uniform, normal nor an exponential distribution. This situation is common to most simulation experiments in general as the class of the steady state distribution is usually unknown. Here, we compare with the theoretical steady state distribution functions, which are derived analytically in Appendix A.4 and Appendix A.5. These distributions are needed during coverage analysis.

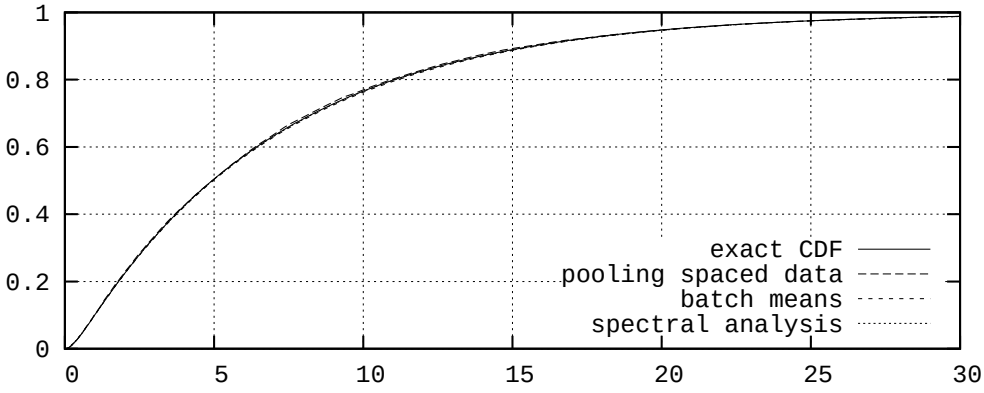
Similar to our experiments with an M/M/1 queue we choose different traffic intensities ρ for the M/E_2/1 queue. Here, μ denotes the service rate of a single stage of the Erlang distribution. Thus, the traffic intensity is given by $\rho = \frac{2\lambda}{\mu}$ in this case. The M/E_2/1 queue has a lower service time variance because its coefficient of variation, i.e. standard deviation divided by mean of the service time, is lower than 1. We set $\lambda = 1$ for all experiments and adjusted $\mu = \{\frac{1}{0.25}, \frac{1}{0.375}, \frac{1}{0.45}\}$ to produce $\rho = \{0.5, 0.75, 0.9\}$. The NOBM approach and spectral analysis use Equation (2.16), whereas the pooling approach uses Equation (2.15). Figure 6.21 shows that the estimated probability distribution functions are nearly identical with the exact distribution. This is true for all methods and for all values of ρ. However, the coverage of the quantile estimates is as expected only for the pooling approach. This can be seen in Figure 6.22, Figure 6.23 and Figure 6.24 for the different values of ρ. The coverage of the NOBM approach and spectral analysis is not as expected at all. Even some quantiles, which are not extreme in sense of a low value $f_X(x)$, show a level of coverage which is far beyond the expected value. This can be explained by the use of Equation (2.16), it is the best choice only for exponential distributions. Because the sample size p is fixed for the NOBM ap-



(a) traffic intensity $\rho = 0.5$

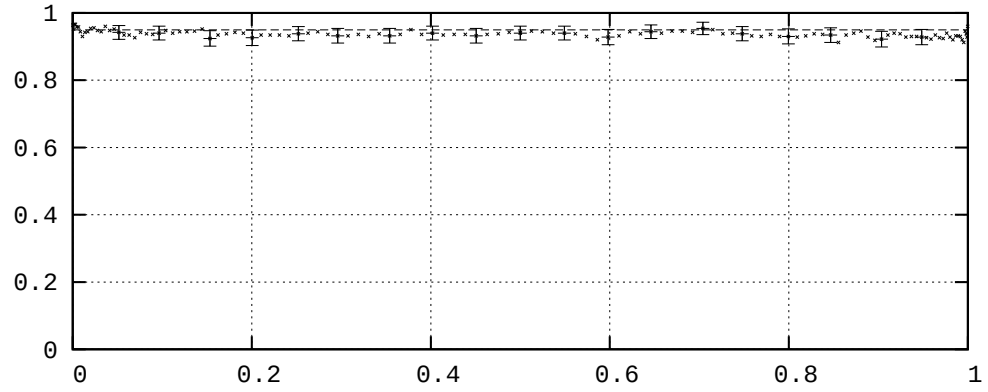


(b) traffic intensity $\rho = 0.75$

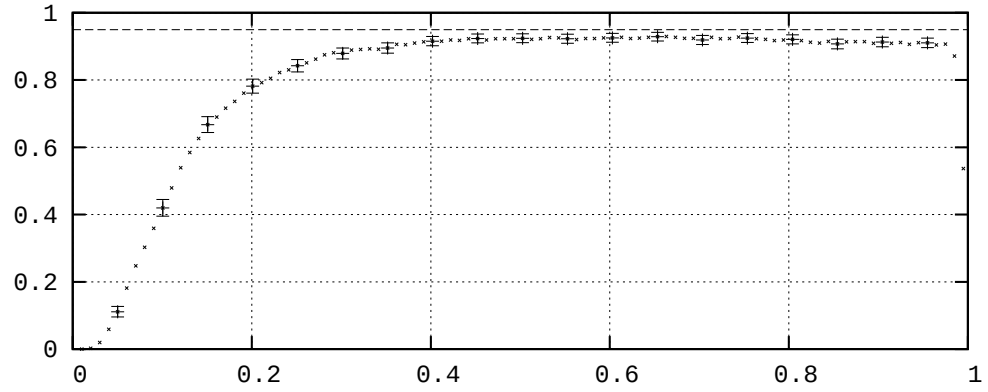


(c) traffic intensity $\rho = 0.9$

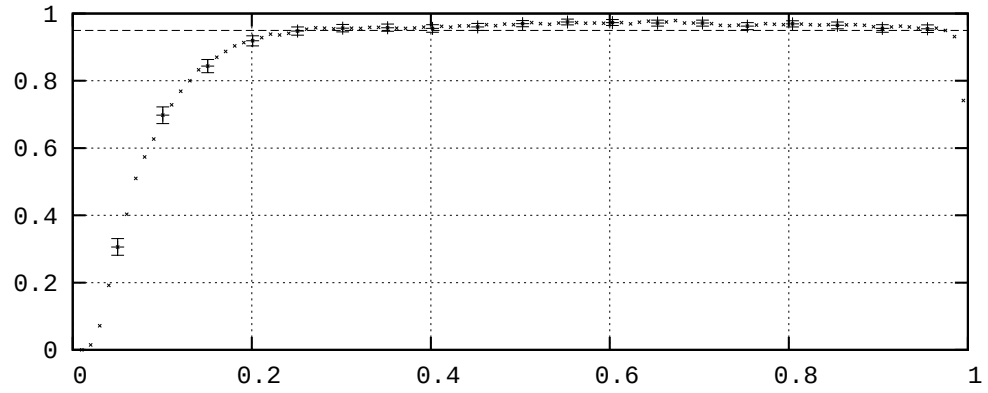
Figure 6.21: Exact and estimated CDFs of the response time of an M/E_2/1 queue with various traffic intensities ρ.



(a) pooling of spaced data

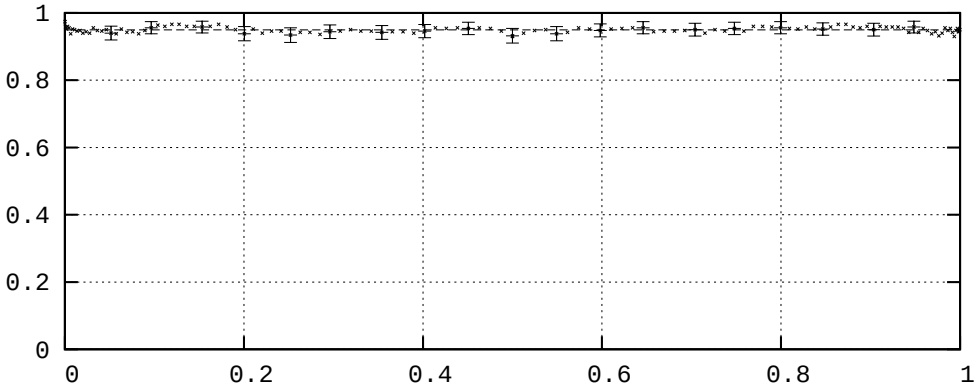


(b) NOBM

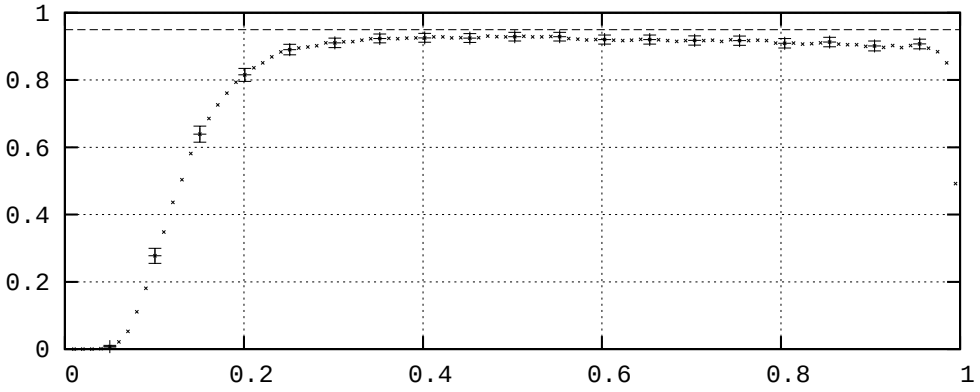


(c) spectral analysis

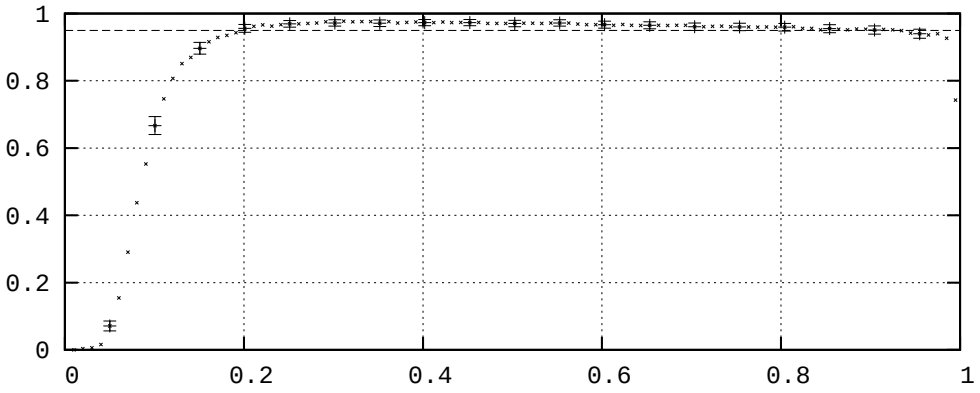
Figure 6.22: Coverage (ordinate) of the q-quantile (abscissa) of the response time of the M/E_2/1 queue with traffic intensity $\rho = 0.5$.



(a) pooling of spaced data

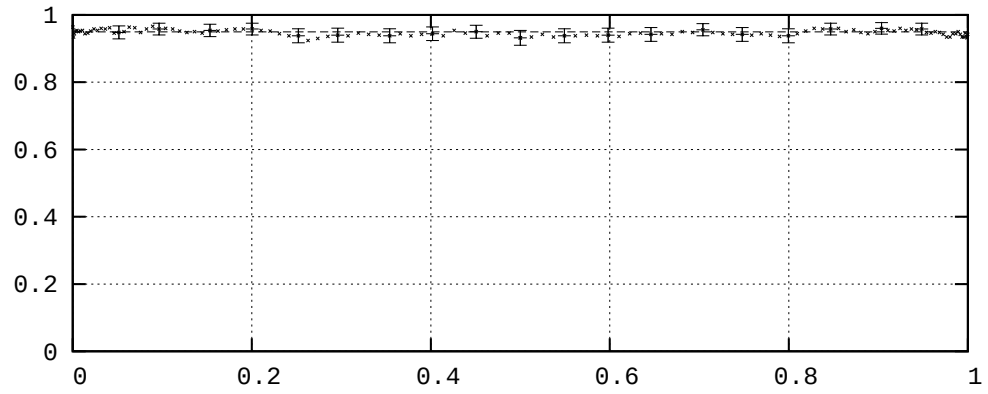


(b) NOBM

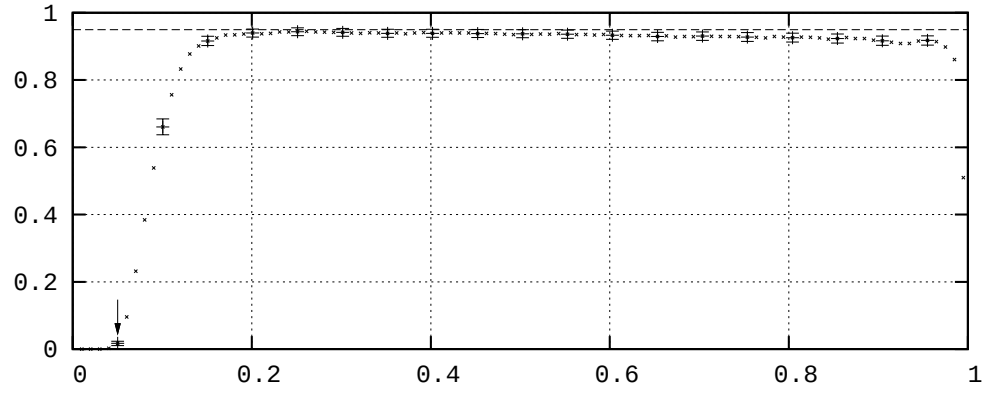


(c) spectral analysis

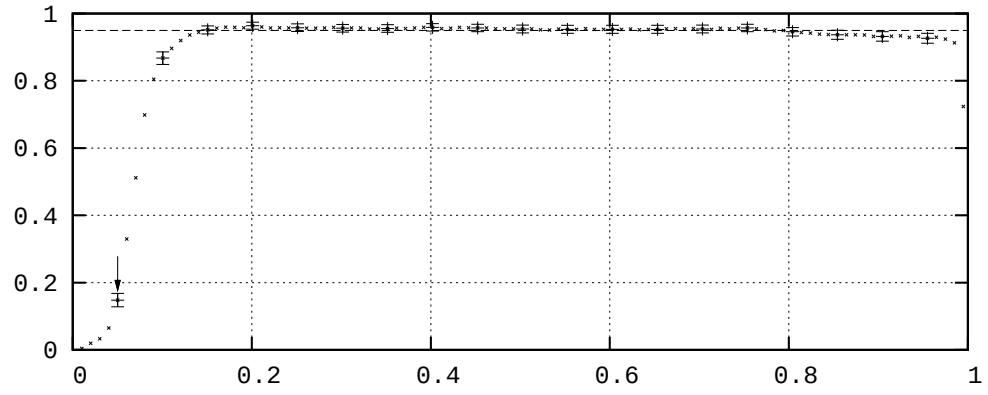
Figure 6.23: Coverage (ordinate) of the q-quantile (abscissa) of the response time of the M/E_2/1 queue with traffic intensity $\rho = 0.75$.



(a) pooling of spaced data



(b) NOBM



(c) spectral analysis

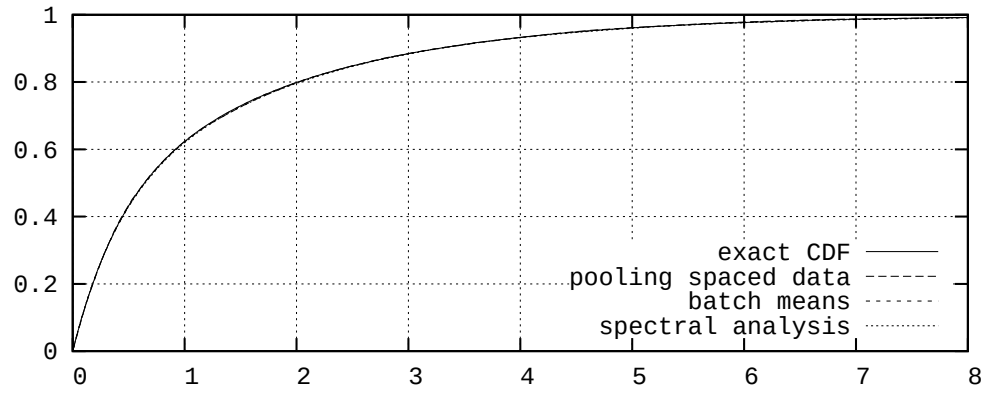
Figure 6.24: Coverage (ordinate) of the q-quantile (abscissa) of the response time of the M/E_2/1 queue with traffic intensity $\rho = 0.9$.

proach and spectral analysis, the approximate properties of Equation (2.16) are not strong enough. The estimate of some quantiles will be biased and we can expect that wherever the given distribution has a different form than the exponential distribution the coverage is poor. However, the form of the estimated distribution is still closer to an exponential distribution than to a uniform or normal distribution. Thus, Equation (2.15) or Equation (2.17) will not provide better results. A typical space size when pooling spaced data is $s = 16$ (for $\rho = 0.5$), $s = 64$ (for $\rho = 0.75$) and $s = 256$ (for $\rho = 0.9$). Typical values of the minimum batch size when using NOBM are $m = 16$ (for $\rho = 0.5$), $m = 128$ (for $\rho = 0.75$) and $m = 512$ (for $\rho = 0.9$). When performing spectral analysis typically batches of size $m = 4$ (for $\rho = 0.5$), $m = 32$ (for $\rho = 0.75$) and $m = 64$ (for $\rho = 0.9$) were used to reduce storage requirements.

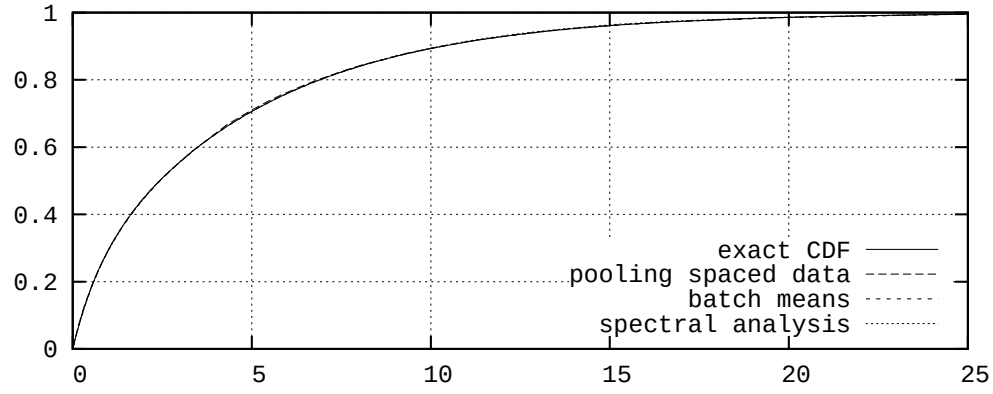
The next set of experiments is done with the M/H_2/1 queue. It has a coefficient of variation of the service time greater then 1. In all our experiments we set $\lambda = 1$, therefore, the traffic intensity is given by $\rho = \frac{p}{\mu_1} + \frac{1-p}{\mu_2}$, where p is the probability of using service rate μ_1 and $1 - p$ is the probability of using service rate μ_2. To obtain $\rho = \{0.5, 0.75, 0.9\}$ we set $p = 0.2113248654$,

$$\begin{aligned} \mu_1 &= \{0.8452994616, 0.5635329745, 0.4696108120\} \quad \text{and} \\ \mu_2 &= \{3.154700538, 2.103133692, 1.752611410\}. \end{aligned}$$

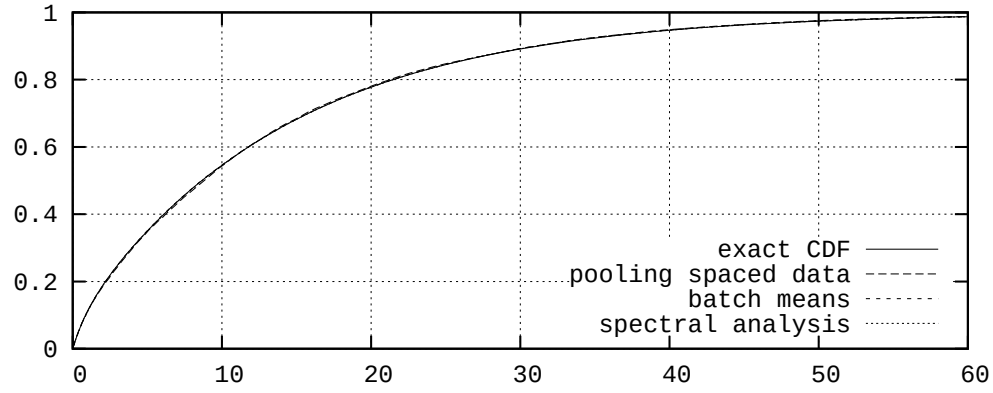
These settings give a squared coefficient of variation of 2 and use the common device of balanced means, i.e. $\frac{p}{\mu_1} = \frac{1-p}{\mu_2}$. As we can see in Figure 6.25 the estimated steady state distributions are nearly indistinguishable from the theoretical distribution for all of the three methods. However, the coverage analysis reveals that the pooling approach delivers the best results because its coverage is always as expected, see Figure 6.26, Figure 6.27 and Figure 6.28. The coverage of the NOBM approach and spectral analysis is poor for some quantiles. This is due to Equation (2.16) and a fixed p, as already pointed out, this equation is the best



(a) traffic intensity $\rho = 0.5$

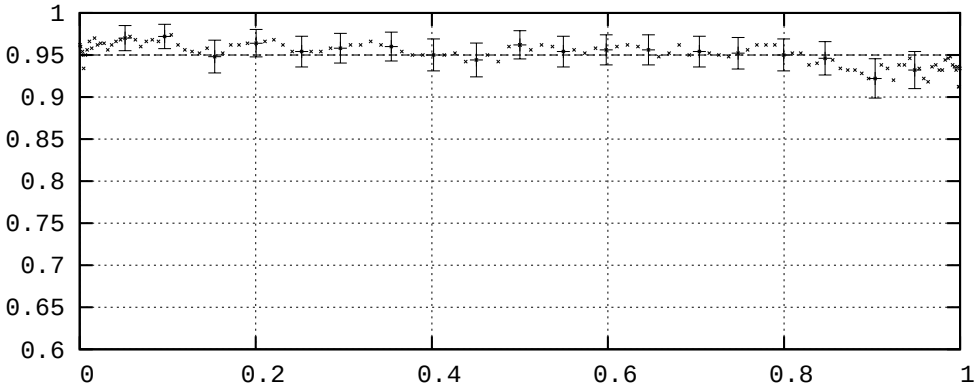


(b) traffic intensity $\rho = 0.75$

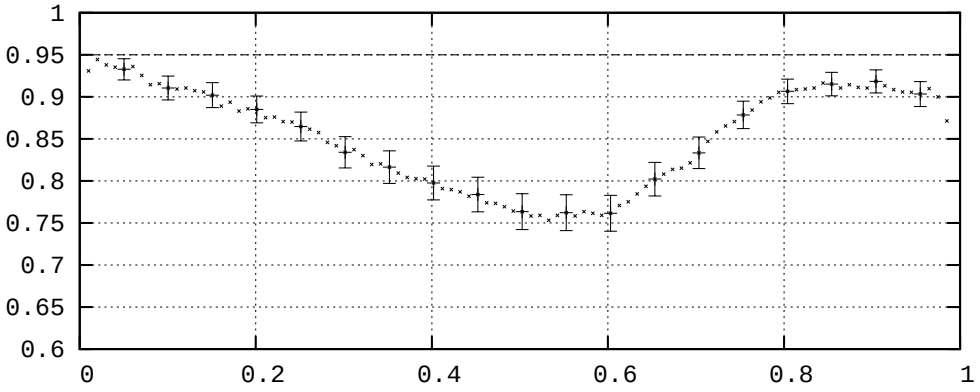


(c) traffic intensity $\rho = 0.9$

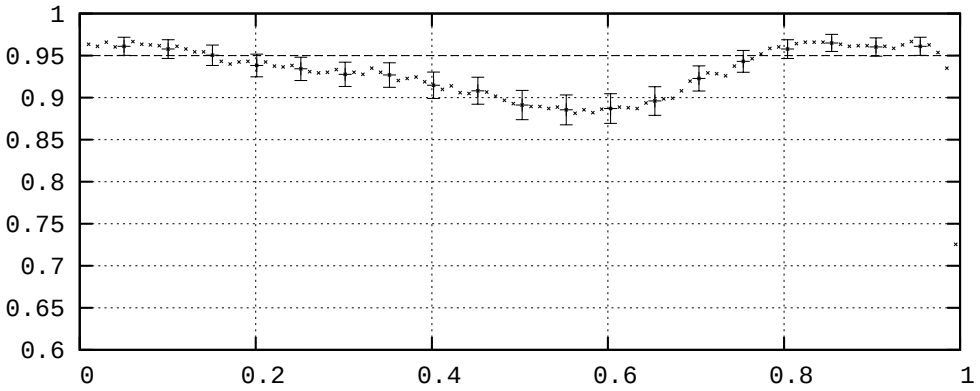
Figure 6.25: Exact and estimated CDFs of the response time of an M/H_2/1 queue with various traffic intensities ρ.



(a) pooling of spaced data

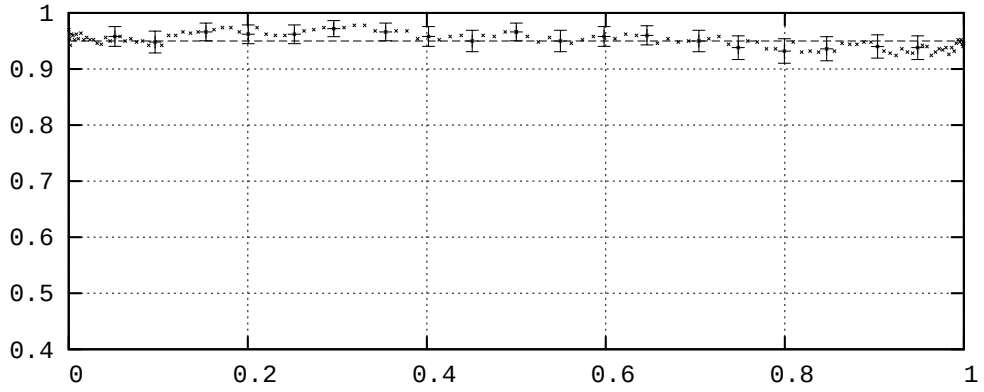


(b) NOBM

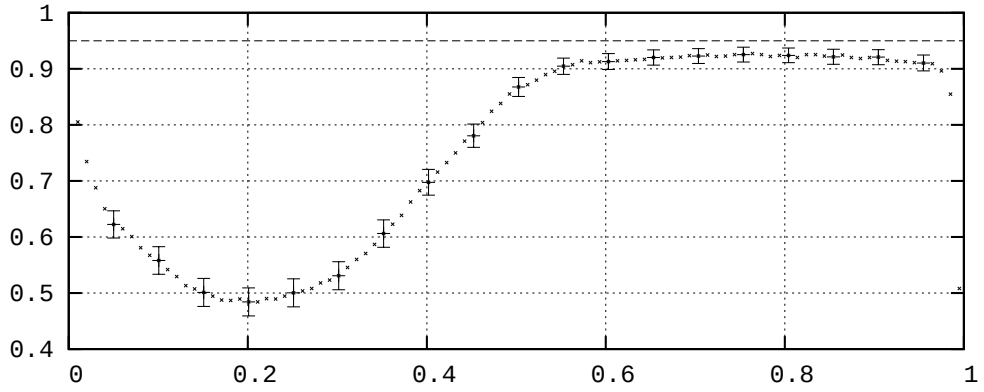


(c) spectral analysis

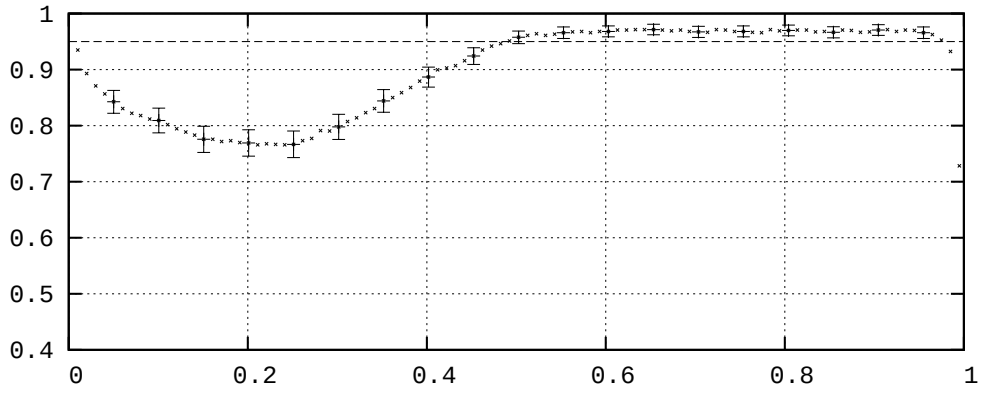
Figure 6.26: Coverage (ordinate) of the q-quantile (abscissa) of the response time of the M/H_2/1 queue with traffic intensity $\rho = 0.5$.



(a) pooling of spaced data

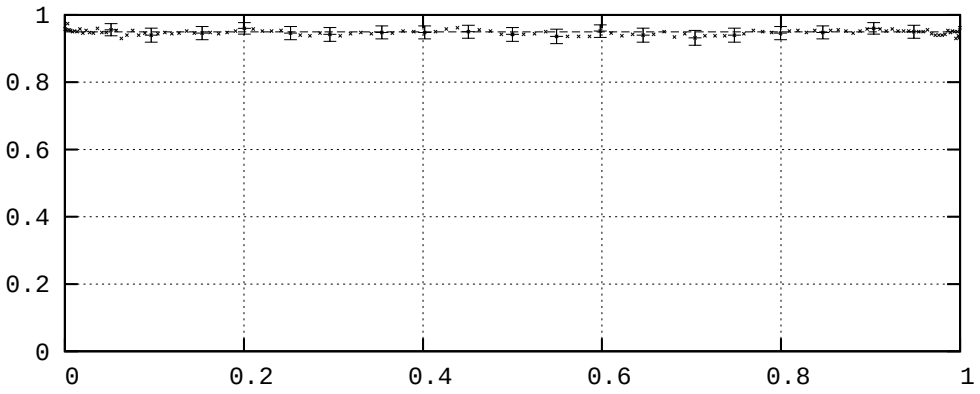


(b) NOBM

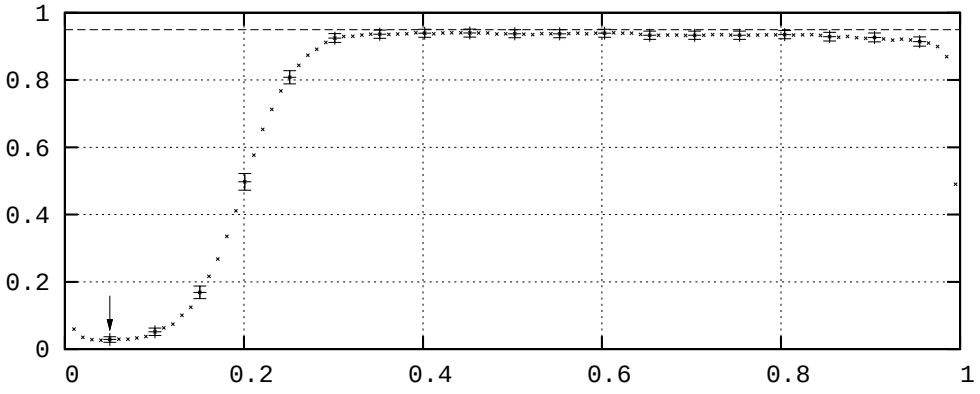


(c) spectral analysis

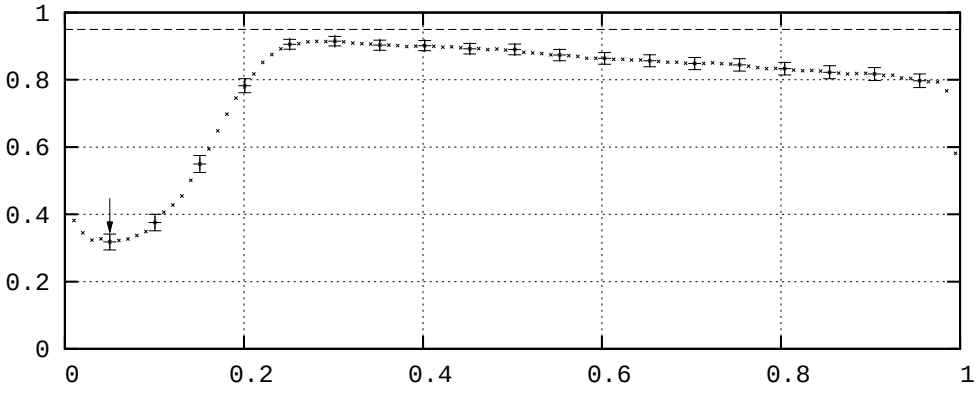
Figure 6.27: Coverage (ordinate) of the q-quantile (abscissa) of the response time of the M/H_2/1 queue with traffic intensity $\rho = 0.75$.



(a) pooling of spaced data



(b) NOBM



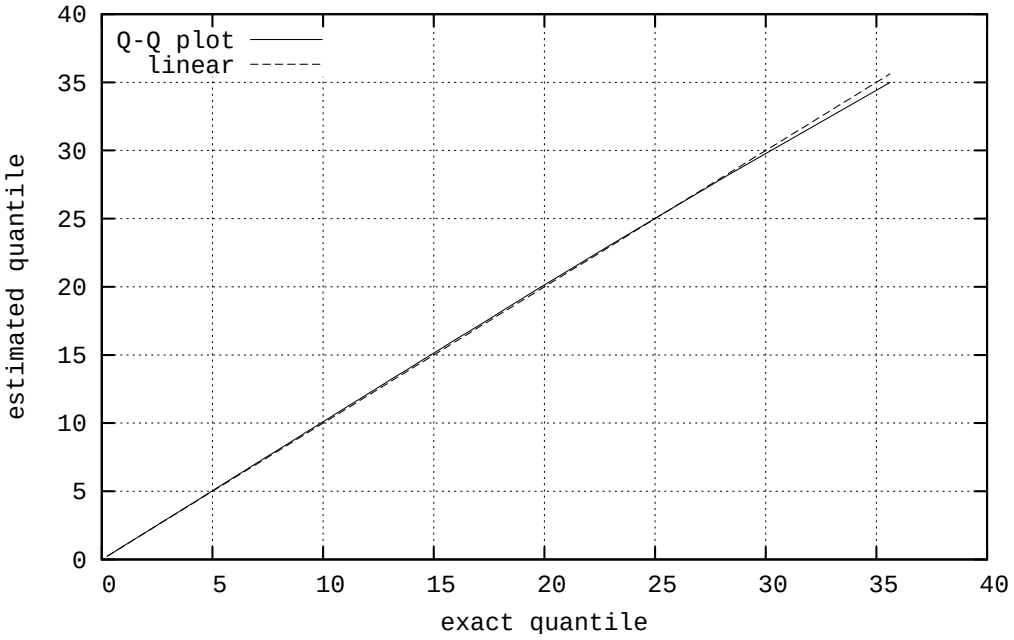
(c) spectral analysis

Figure 6.28: Coverage (ordinate) of the q-quantile (abscissa) of the response time of the M/H_2/1 queue with traffic intensity $\rho = 0.9$.

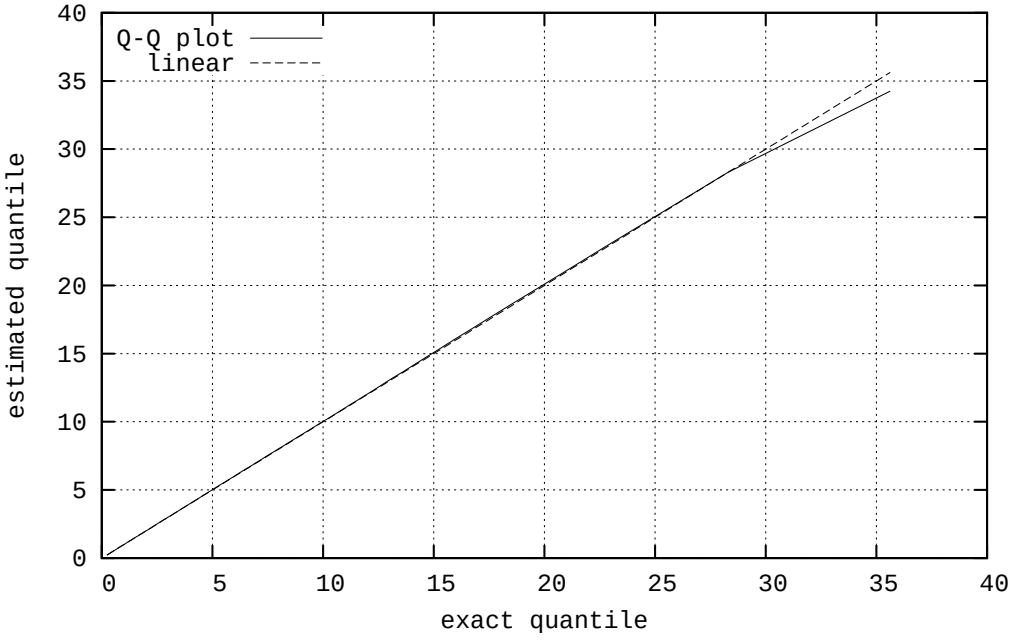
choice only for exponential distributions.

A typical space size when pooling spaced data is $s = 16$ (for $\rho = 0.5$), $s = 128$ (for $\rho = 0.75$) and $s = 1024$ (for $\rho = 0.9$). Typical values of the minimum batch size when using NOBM are $m = 32$ (for $\rho = 0.5$), $m = 512$ (for $\rho = 0.75$) and $m = 1024$ (for $\rho = 0.9$). When performing spectral analysis typically batches of size $m = 4$ (for $\rho = 0.5$), $m = 32$ (for $\rho = 0.75$) and $m = 128$ (for $\rho = 0.9$) were used to reduce storage requirements. It is quite surprising that the coverage of the NOBM approach and spectral analysis is so poor, despite the fact that the estimated distributions are nearly indistinguishable from the exact distribution. For this reason we depicted the Q-Q plot for the M/E_2/1 queue in Figure 6.29 and for the M/H_2/1 queue in Figure 6.30 with $\rho = 0.9$. In our Q-Q plots the estimated quantiles (ordinate) are shown in dependence of the exact quantiles (abscissa). In the best case the resulting curve should be linear. In all our Q-Q plots we can see a nearly linear curve, except extreme quantiles differ from the linear form. This indicates again that the estimated distribution is nearly indistinguishable from the exact distribution. However, the reason why the coverage is quite poor can be understood by drawing the distribution function of the quantile estimate itself.

In Figure 6.31 we show results of the M/E_2/1 queue with $\rho = 0.9$, where the empirical CDF of the 5th order statistic is shown, which represents a quantile with poor coverage, see arrows in Figures 6.24(b) and 6.24(c). According to Equation (2.16) this order statistic represents the quantile at $q = \frac{i}{p+\frac{1}{2}} = 0.0502513$, where $i = 5$ and $p = 99$. Furthermore, according to the exact distribution $F_X^{-1}(q) = 0.609874$ (dashed arrows in Figures 6.31(a) and 6.31(b)). The averages of all estimates (bold arrows in Figures 6.31(a) and 6.31(b)) are $\hat{F}_X^{-1}(q) = 0.601677$ and $\hat{F}_X^{-1}(q) = 0.601701$ for the NOBM approach and spectral analysis, respectively. The difference $|F_X^{-1}(q) - \hat{F}_X^{-1}(q)|$ is in both cases smaller than 0.0082, which is a really small value and is too small to be detected by a visual inspection of the graph of the CDF in Figure 6.21(c). In Figure 6.31 we can see

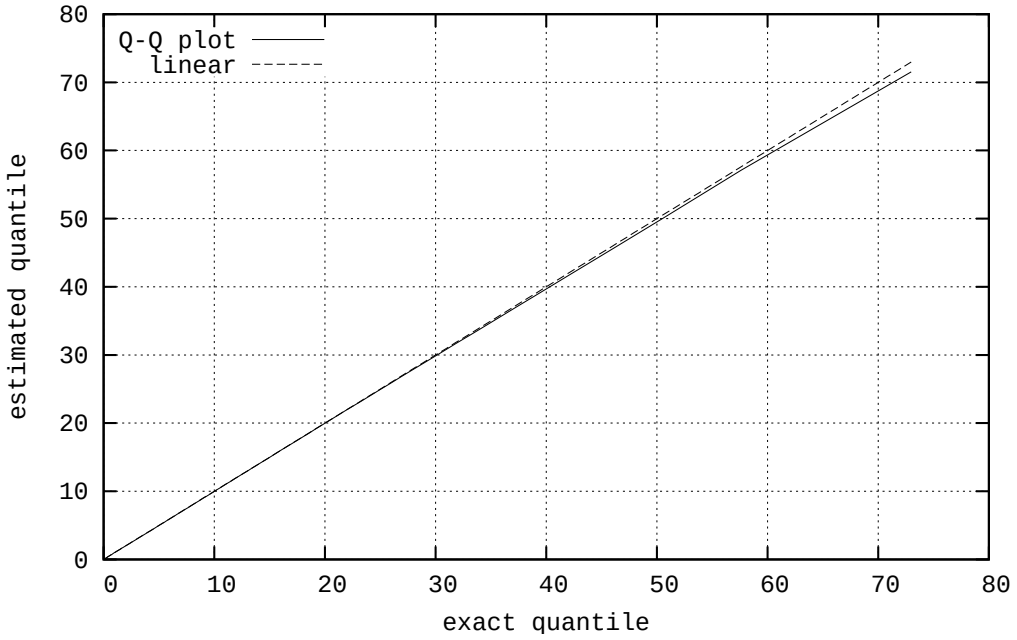


(a) NOBM

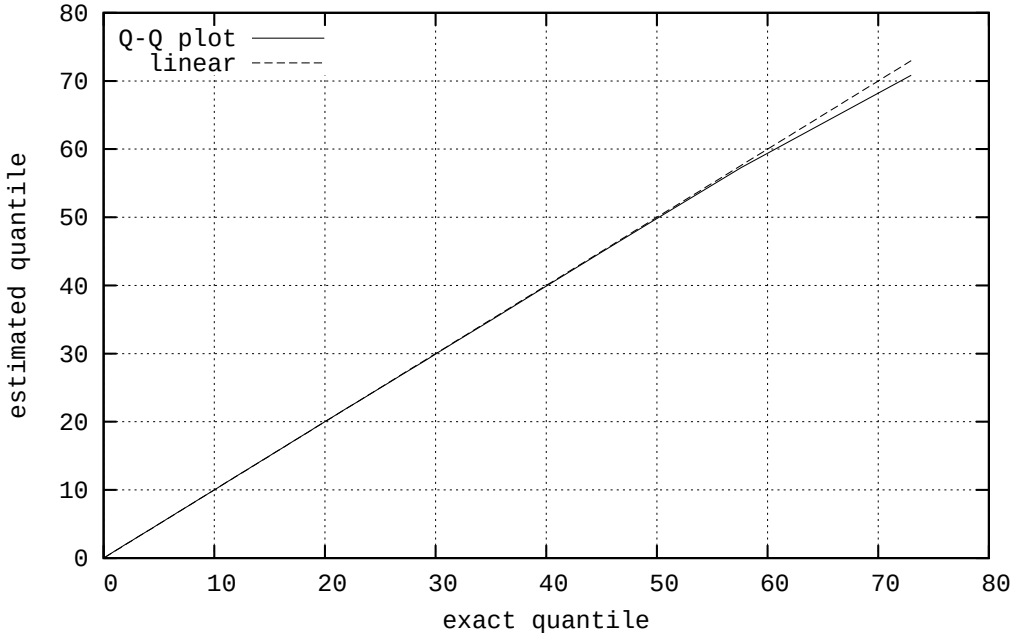


(b) spectral analysis

Figure 6.29: QQ-plot of the exact and estimated CDF of the response time of an M/E_2/1 queue with traffic intensity $\rho = 0.9$.

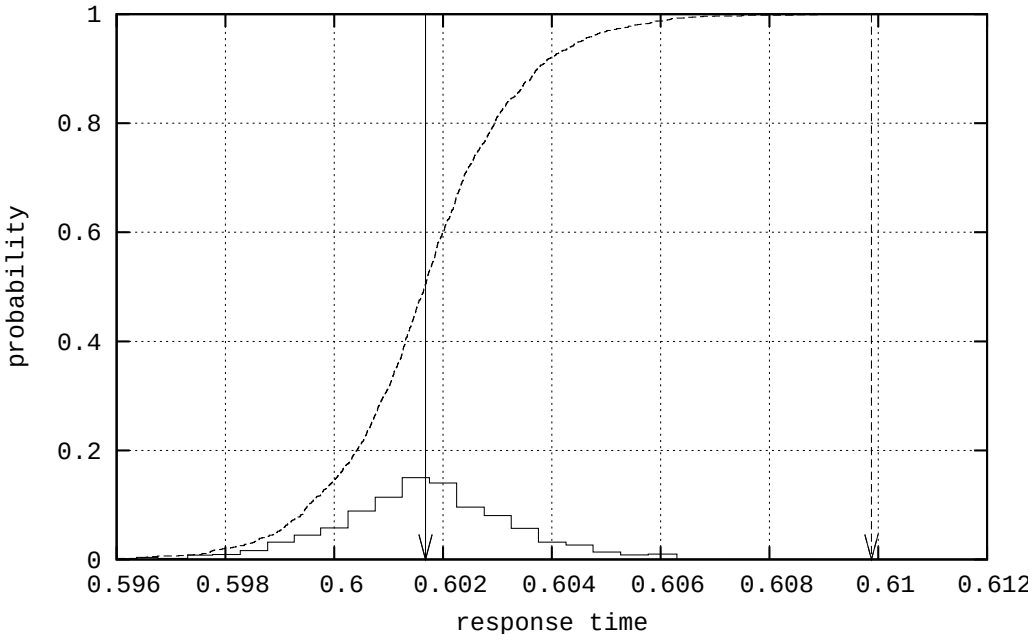


(a) NOBM

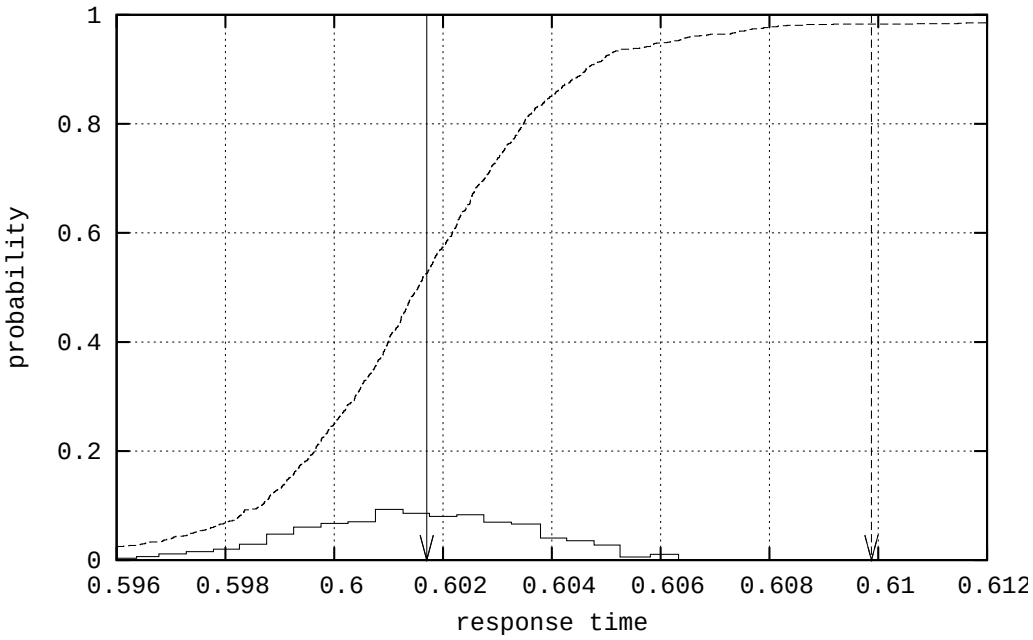


(b) spectral analysis

Figure 6.30: QQ-plot of the exact and estimated CDF of the response time of an M/H_2/1 queue with traffic intensity $\rho = 0.9$.

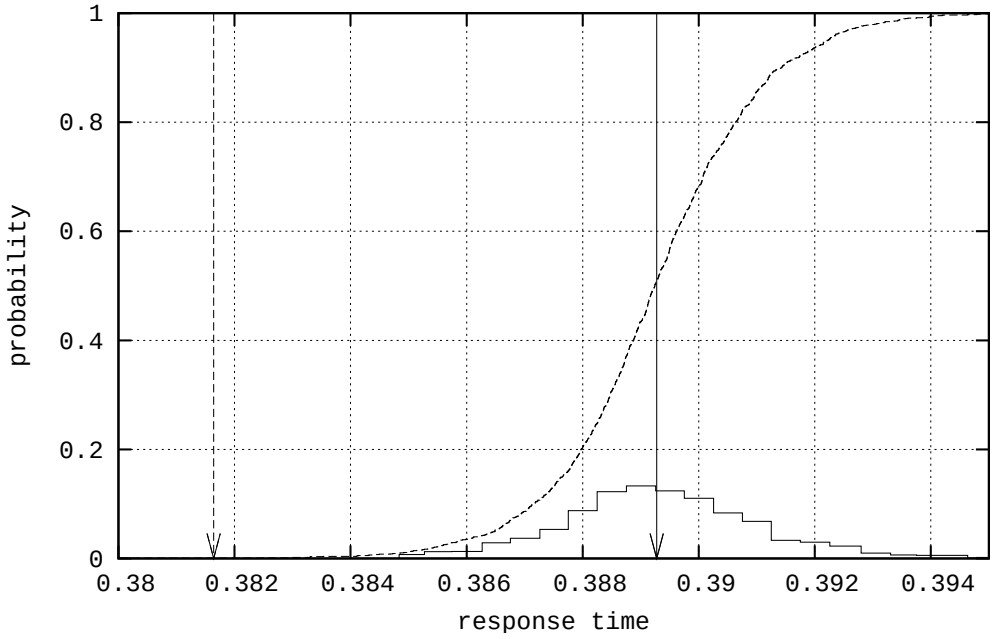


(a) NOBM

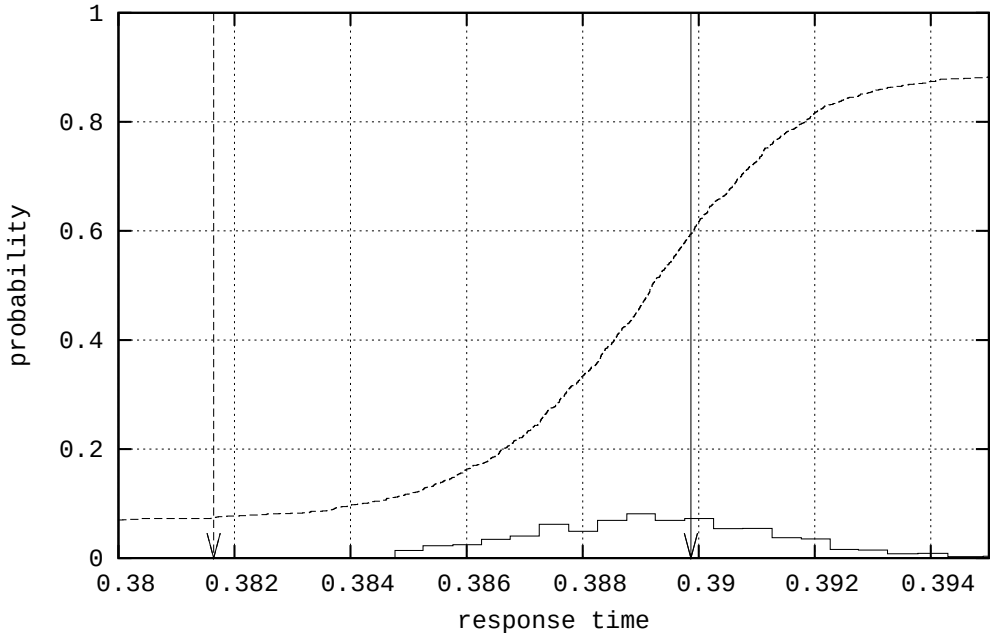


(b) spectral analysis

Figure 6.31: Empirical CDF of the 5th order statistic ($p = 99$) for an M/E_2/1 queue with traffic intensity $\rho = 0.9$.



(a) NOBM



(b) spectral analysis

Figure 6.32: Empirical CDF of the 5th order statistic ($p = 99$) for an M/H_2/1 queue with traffic intensity $\rho = 0.9$.

that the distribution of the estimates is very focused. For example the observed range of the distribution of the estimates of the NOBM approach is smaller than 0.14, which is tiny compared to the observed range of the underlying distribution of the response time, which is greater than 30. The confidence intervals of the estimates are similarly small. This is why the coverage is quite poor in this example. For the 5th order statistic in the example of the M/H_2/1 queue with $\rho = 0.9$, see arrows in Figures 6.28(b) and 6.28(c), we can draw similar conclusions. Here, we expect $F_X^{-1}(q) = 0.381642$ (dashed arrows in Figures 6.32(a) and 6.32(b)). The average of all estimates (bold arrows in Figures 6.32(a) and 6.32(b)) are $\hat{F}_X^{-1}(q) = 0.389277$ and $\hat{F}_X^{-1}(q) = 0.389866$ for the NOBM approach and spectral analysis, respectively. The differences $|F_X^{-1}(q) - \hat{F}_X^{-1}(q)| < 0.0083$ are very similar and really small. Again, the bad coverage can be explained by the focused distribution of the estimates.

6.4.5 Verification for the M/E_2/1 Queue

In the previous section we demonstrated that bad coverage of the final quantile estimates appears for methods, which are based on mean value estimation. This indicates that bad coverage is caused by the bias, see Equation (2.2), of the point estimate and not by the size of the interval estimate. The estimated halfwidth of the NOBM approach or spectral analysis does not seem to be the source of the problem. One reason for the bias is that Equation (2.16) is the best choice for a true exponential distribution only. The second reason is that the methods, which are derived from mean value analysis, operate with a fixed sample size p. This might be not sufficient for some quantiles. The assumption of a large p in Corollary 6.2.2 is violated.

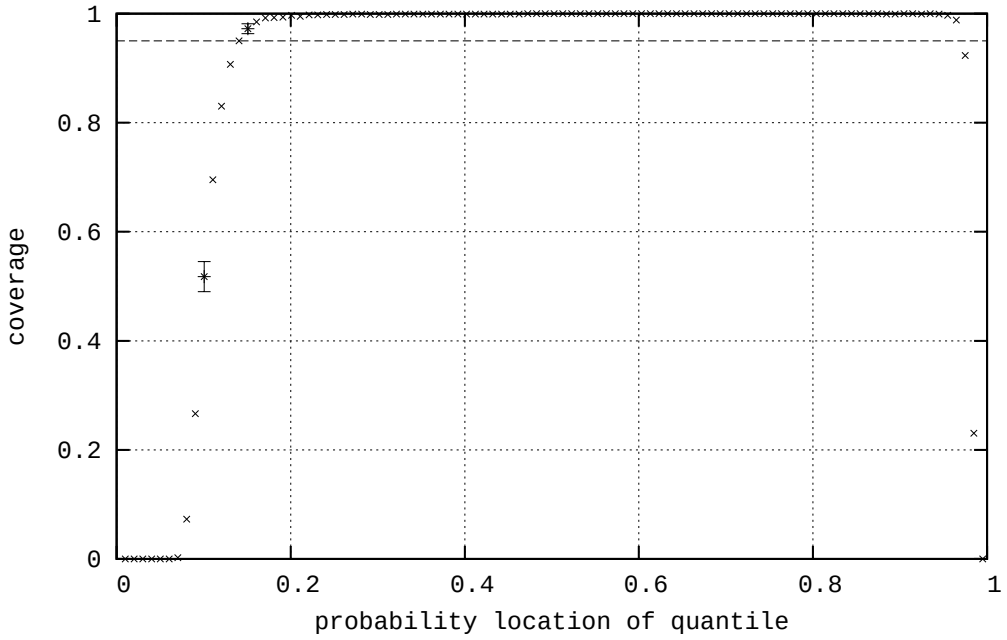
To clearly show that the bad coverage is cause by the point estimator's bias we perform simulation experiment with a basic process, as in Section 6.4.2. Here, the distribution of each X_i is the exact steady state distribution of the M/E_2/1 queue

for $\rho = 0.9$ and there is no autocorrelation present. We set all parameters as described in the previous section and forced the NOBM approach to use the same batch size as in the autocorrelated case. If the bad coverage is caused by the point estimator's bias, we expect that quantile estimates with negligible bias will have a coverage close to 1 because the batch size is higher than necessary. We expect that coverage of biased estimates will be close to 0 and even worse than before because the distribution of the estimates of the basic process are more focused due to better statistical properties of the output process.

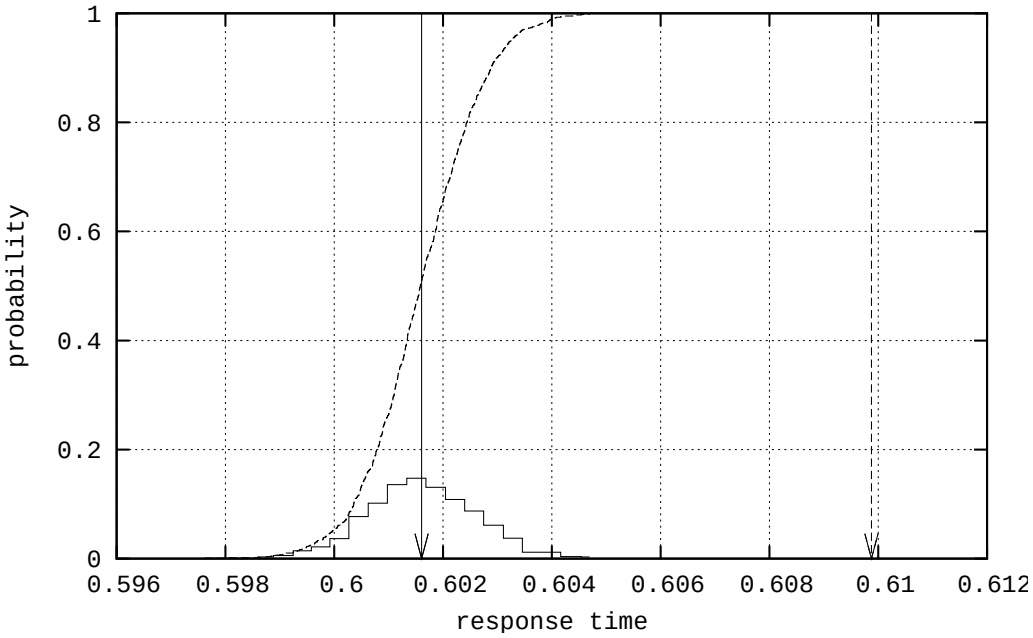
The simulation results for spectral analysis are shown in Figure 6.33. We can see that the form of the curve in Figure 6.33(a) is similar to the curve in Figure 6.24(c). Quantile estimates without bias show a coverage of 1, because here the output data is uncorrelated but the used batch sizes are valid for autocorrelated output data. The coverage of the estimate of the 5th order statistic is close to 0. In Figure 6.33(b) we can see that the distribution of these estimates is even more focused than in Figure 6.31(b). According to Equation (2.16), the 5th order statistic represents the quantile $q = \frac{i}{p+\frac{1}{2}} = 0.0502513$ with $F_X^{-1}(q) = 0.609874$. The average of all quantile estimates at q is $\hat{F}_X^{-1}(q) = 0.601605$. Therefore, the bias $F_X^{-1}(q) - \hat{F}_X^{-1}(q) \approx 0.0082$ is comparable to the experiments with the M/E_2/1 queue. This result supports our previous statement that the estimator defined in Equation (2.16) is not accurate because the number of replications p should not be fixed.

Coverage analysis is commonly used to test the quality of an interval estimate. In the case of an M/E_2/1 or M/H_2/1 queue the point estimate is already biased. Coverage analysis might not be an appropriate measure in this case. However, the results in this section show that the bad coverage is caused by not entirely fulfilling the assumptions of Corollary 6.2.2. The estimation of the variance by NOBM or spectral analysis is clearly not the source of error.

On basis of the experiments with the M/E_2/1 and M/H_2/1 queues we can con-



(a) Coverage (ordinate) of the q-quantile (abscissa).



(b) Empirical CDF of the 5th order statistic ($p = 99$).

Figure 6.33: Results for the basic process distributed as the steady state distribution of the M/E_2/1 queue for $\rho = 0.9$.

clude that the results of the pooling approach are the best. This is due to the growing sample size of the pool of observations. The estimates of the NOBM approach and spectral analysis are biased because p is a constant value and Equation (2.16) is not the best choice. It might be possible to find an even more specialised transformation. However, we are interested in a general solution for automated simulation, thus, this is not regarded any further. The NOBM approach and spectral analysis cannot benefit from a growing sample size because just the mean of the rank statistics is becoming more precise. Each rank provides a biased estimate because p is constant and too small.

6.5 Limits and Conclusion

All experiments in this section clearly indicate that the quantile estimator, which is based on pooling spaced data, see Section 6.3, has the best statistical properties. The growing size of the pool of spaced observations guarantees that the bias due to Equation (2.15) is small. The detection of the truncation point assures that the data is identically distributed and the selection of the space size transforms the data into a sample of independent observations. The coverage of the confidence intervals of estimated quantiles is as expected, even for extreme quantiles. The form of the underlying probability distribution function is not significant, neither is the form of the transient behaviour or the degree of autocorrelation. The underlying CDF can be approximated by drawing a curve through the estimated quantiles. This curve is nearly indistinguishable from the expected curve of the underlying distribution. The parameterisation of this method does not contain any critical parameters. A disadvantage is the growing size of the pool of spaced observations. However, this size is just depending on the parameterisation and not on properties of the output process. All in all, this method should be preferred because of its statistically reliable results.

The advantage of the quantile estimators, which are based on NOBM and spec-

tral analysis, see Section 6.2, is the guarantee of constant storage requirements. Both methods operate on a constant number of batches. The drawback is that the assumptions of Theorem 6.2.1 cannot be fulfilled completely. p is assumed to be large to assure an unbiased estimate. Our experience show that a common setting of $p = 99$ is too small for extreme quantiles. This can be compensated by choosing Equation (2.15), Equation (2.16) or Equation (2.17) depending on the underlying distribution. However, if none of the three equations is adequate for the underlying distribution some estimated quantiles will be biased. Note that NOBM usually chooses the largest batch size where as spectral analysis usually operates on the smallest batch size because in this case no independence between batch statistics is assumed.

Chapter 7

Conclusions and Future Work

In this chapter final conclusions about results of this research work are made and possible future work is pointed out.

7.1 Conclusions

In the introductory chapters we pointed out that there are many performance measures based on different characteristics of output processes, as stated in Chapter 2, which are potentially of interest. The most popular characteristics are expectations of random values, which are usually easiest to estimate. Currently, sequential simulation output analysis is basically confined to mean value analysis only. However, the information gain of mean value analysis is limited, because it can only provide information about a common or average behaviour of the analysed system. In contrast to this stands quantile analysis, which can provide information about the probability of a certain behaviour. Furthermore, a set of carefully chosen quantiles can provide an impression of the whole probability distribution of a given random value.

The main difficulties of quantile estimation in simulation output analysis is that typical output data from simulation is not identically distributed, due to an initial transient, and autocorrelated. The initial transient possibly leads to a bias of point estimates and the correlation potentially leads to a wrong size of interval

estimates. We demonstrated in Chapter 3 that collecting output data from synchronous independent replications can greatly assist the estimation of quantiles, because the difficulties of quantile analysis, the initial transient and autocorrelation, can be targeted in a new way. Many difficulties of quantile estimation could be resolved by applying synchronous independent replications, which are used in all of the new methods we discussed in Sections 4, 5 and 6. We could show that the use of replications does not only offer a speedup but different statistical methods can be applied on the output data. This is the key advantage we used to introduce new estimation methods.

A method that analyses the time evolution of a set of quantiles is introduced in Chapter 4. This provides an insight into the dynamics of the simulated model. The set of quantiles is chosen automatically to produce disjoint confidence intervals so that high correlation between quantile estimates of one observation index is avoided. The estimation method is robust and can be used for many kinds of time dependent behaviour. The degree of correlation of the output process does not influence the quality of the estimates substantially. The size of the quantile's confidence intervals in the probability domain can be calculated directly. The size of the quantile's confidence intervals in the domain of the measure of interest can be calculated during the simulation experiment. Therefore, the error can be controlled to meet a predefined threshold.

The use of homogeneity tests opens a new class of truncation point detection methods, as discussed in Chapter 5. This new class of methods is based on the convergence of the probability distribution towards the steady state distribution. Therefore, these methods are more powerful than other known methods, which are based on weaker criteria of convergence, such as convergence of the mean. These methods can be used on a broader class of output processes and reduce the bias in subsequent estimators. The homogeneity tests are embedded into different algorithmic approaches, which are compared by several examples. To obtain a

sufficiently large random sample for the homogeneity tests not less than 30 replications should be used. The new truncation point detection methods are tested for many time dependent processes, including processes which are non-stationary, periodic, non-monotone, have a transient mean, a transient variance or converge very slow. We tested the new methods for continuous probability distributions only to avoid having to adjust homogeneity test statistics for ties in random samples from discrete probability distributions. Only parameters, which are model independent, have to be set by the user. All parameters, which require previous knowledge of the model, are chosen automatically on basis of the simulation output data.

A method that estimates a set of quantiles of the steady state distribution is introduced in Chapter 6. We showed that the quantile estimates are approximately unbiased. The experimental coverage of the quantile's interval estimates is statistically equal to the expected coverage. This is true for all quantiles, especially for those in areas of low probability. However, we do not recommend the use of these methods to estimate extreme quantiles. In this case, rare event simulation would be more advisable to reduce the run time of the simulation. Again, the set of quantiles is chosen automatically with disjoint confidence intervals so that high correlations between quantile estimates is avoided. The method is tested for many kinds of continuous distributions and various autocorrelation structures. The parameterisation of the method does not require previous knowledge of the model or its output data. We demonstrated that the set of quantiles can be used to estimate the underlying cumulative distribution function.

All of the discussed methods can be used in automated and sequential analysis. No parameters have to be specified which require previous knowledge of the model or the output data. Point estimates are always given with a confidence interval. The size of the confidence interval is reduced until it meets the sequential stopping criterion. This guarantees valid final estimates with small statistical error.

As already pointed out in Chapter 1, we have focused on continuous probability distributions only. When analysing discrete probability distributions the ties in random samples have to be considered. This is important for the definition of quantiles as well as for homogeneity test statistics. However, the extension to discrete distributions should be straightforward, if laborious. Furthermore, we assumed that lower moments of the analysed probability distribution are finite. This limitation could be important when analysing heavy tailed distributions.

The examples in the various chapters show that quantile estimation provides a deep insight into the dynamics of the simulation experiment and the model's behaviour. Quantile estimates can be used to analyse evolution over time as well as to analyse the steady state distribution of arbitrary performance measures.

7.2 Future Work

We already pointed out that implementations of proposed estimation methods have focused on analysis of continuous distribution functions. However, analysis of discrete measures, such as population or queue length, could be of interest to the analyst as well. In future work the implementations of proposed estimation methods could be extended to cover discrete measures by adjusting test statistics for e.g. homogeneity test. The detection of ties, i.e. identical values, within observed random samples could be used to automatically distinguish between the continuous and the discrete case. Because of reasons of efficiency such a detection method has to be based on a carefully chosen algorithmic approach. It might not be adequate to compare each new collected observation with all previously collected observations when searching for identical values. The CDF of a discrete measure is usually a step function. This means that the common definition of quantiles, see Equation (2.12), can lead to multiple quantiles with the same location. We believe that this will not be a major issue when estimating quantiles, however, the depiction of the CDF based on a linear interpolation between

quantiles might lead to inappropriate results.

We confined the validation of the proposed estimation methods to well behaved random variables. This excludes for example heavy tailed distributions and distributions with undefined moments. In an extreme case even the mean, the first moment, could be infinite. Because we have not examined random variables with these kind of properties we are not able to give a statement about the performance of the proposed estimation methods used for these examples. In future work detailed properties of not well behaved random variables could be identified which guarantee statistically valid quantile estimates. On the other hand cases could be identified where quantile estimation is not possible at all, for example if the value of a quantile is infinite. We expect that the implementation of estimation methods has to be extended and maybe even the quantile estimator itself has to be adjusted. This task includes further research work as well as additional experimental studies.

In the literature on quantile estimation it is mentioned that the estimation of extreme quantiles, i.e. for $q \to 0$ and $q \to 1$, is more difficult than the estimation of e.g. the median, see for example [71-HL84]. This confirms to our experience in the case of normally distributed random variables. However, we extend this statement by pointing out that quantile estimation is more difficult in areas where the probability density is low, i.e. $f_X(x) \to 0$. The aim of this research work is to provide an estimate of the whole distribution function, thus, areas of low probability density do not receive special treatment. However, the analyst could be interested especially in information about these areas, for example to estimate the probability of an unlikely buffer overflow. In future work the proposed estimation methods could be extended to treat areas of low probability density separately. A possible extension of the proposed estimation methods is to incorporate them with rare event simulation. Rare event simulation is a promising technique to implement this task because in rare event simulation the speed of producing relevant

observations is increased by focusing on system states with low probability.

In the MRIP scenario, see Section 3.3, analysis of output data is distributed to replications. In this sense the MRIP scenario uses replications of a simulation experiment, which includes production and analysis of local observations. This provides maximum speedup because every computation engine is able to run at its own speed. Local results of each replication can be combined to a global estimate. When estimating quantiles it is difficult to derive local estimates, therefore, we applied synchronous data collection, see Section 3.4, and thus avoid the need for local estimates. In consequence, all computation engines run only as fast as the slowest engine. A future task is to investigate possible speedup by integrating our approach into the MRIP scenario. This could be done by regarding the cluster of all replications of our simulation experiment as one replication of the MRIP scenario. By using many clusters, each providing local estimates, a speedup could possibly be achieved. However, it has to be investigated if global estimates of many small clusters are of the same statistical quality as estimates provided by one large cluster, because the number of replications per cluster effects e.g. the performance of the truncation point detection method of Chapter 5.

Appendix A

Appendix

A.1 Application of Median Confidence Intervals

In Section 6.2.2 and Section 6.3 we pointed out that usually the estimation of an appropriate batch size m is the difficulty of batching methods. Here, our purpose is to estimate the overall batch size m for p independent replications. We would like to apply median confidence intervals to calculate a critical value for Pearson's correlation coefficient. A median confidence interval is a special case of min-max confidence intervals, see [127-Str04]. Let $x_1, x_2, \ldots, x_n$ represent observations collected during a steady state simulation run of length n. And let $X_1, X_2, \ldots, X_n$ denote the corresponding random variables, so that Equation (2.5) applies but not necessarily Equation (2.4). Let $\hat{\Theta}$ be an estimate of an arbitrary parameter Θ we analyse during this simulation run, with CDF $F_{\hat{\Theta}}(x)$. By performing p independent replications we receive $\hat{\Theta}_1, \hat{\Theta}_2, \ldots, \hat{\Theta}_p$ independent estimates of Θ. Let $\hat{\Theta}^{\min} = \min(\hat{\Theta}_i | 1 \leq i \leq p)$ and $\hat{\Theta}^{\max} = \max(\hat{\Theta}_i | 1 \leq i \leq p)$ denote the two extreme values of that sample. As shown in [127-Str04], the min-max confidence interval $[\hat{\Theta}^{\min}, \hat{\Theta}^{\max})$, including $\hat{\Theta}^{\min}$ but excluding $\hat{\Theta}^{\max}$, has the confidence level

$$\Pr\left[\hat{\Theta}^{\min} \leq \Theta < \hat{\Theta}^{\max}\right] = 1 - (F_{\hat{\Theta}}(\Theta))^p - (1 - F_{\hat{\Theta}}(\Theta))^p, \qquad \text{(A.1)}$$

dependent on Θ, which is the characteristic of interest. The difficulty of the min-max confidence interval is to determine $F_{\hat{\Theta}}(\Theta)$. If $F_{\hat{\Theta}}(\Theta) = 0.5$ the min-max confidence interval is called the median confidence interval.

Pearson's correlation coefficient is defined by

$$\frac{\mathrm{Cov}\left[X_0, X_1\right]}{\sqrt{\mathrm{Var}\left[X_0\right]\mathrm{Var}\left[X_1\right]}} \tag{A.2}$$

for two arbitrary random variables X_0 and X_1. Here, let Pearson's correlation coefficient be

$$\hat{r}^{(p)}(P_1) = \frac{\sum_{i=1}^{n_b-1}(s_{j,i}(m) - \bar{s}_{j,0}(m))(s_{j,i+1} - \bar{s}_{j,1}(m))}{\sqrt{(\sum_{i=1}^{n_b-1}(s_{j,i}(m) - \bar{s}_{j,0}(m))^2)(\sum_{i=1}^{n_b-1}(s_{j,i+1} - \bar{s}_{j,1}(m))^2)}} \tag{A.3}$$

of the original lag-1 paired batch statistics $\{(s_{j,i}(m); s_{j,i+1}(m))\}_{i=1}^{n_b-1}$, where n_b is the number of batches and $s_{j,i}(m)$ is an arbitrary statistic calculated of the subsequence $x_{j,(i-1)m+1}$, $x_{j,(i-1)m+2}$, ..., $x_{j,(i-1)m+m}$ and where $\bar{s}_{j,0}(m)$ resp. $\bar{s}_{j,1}(m)$ is the average of $\{s_{j,i}(m)\}_{i=1}^{n_b-1}$ resp. $\{s_{j,i+1}(m)\}_{i=1}^{n_b-1}$. Let P_1 denote the original, i.e. unpermutated, data. And let $\hat{r}^{(p)}(P_k)$ be Pearson's correlation coefficient for the lag-1 paired data of the kth permutation of $\{s_{j,i}(m)\}_{i=1}^{n_b}$ with $2 \leq k \leq (n_b!)$. Pearson's correlation coefficient is probably the most common correlation coefficient, however, other correlation coefficients could be used as well. For example Spearman's correlation coefficient could be used to reduce the influence of outliers because it is based on ranks. We exclusively focus on the lag-1 paired data and disregard higher lags as it is done in [135-WW43] and [84-LC79]. We establish a median confidence interval for $\hat{\Theta} = \hat{r}^{(p)}(P_1)$ under the assumption of $\Theta = 0$. In [107-Pit37] the first four moments of Pearson's correlation coefficient are derived. Here, the first and the third moment are of special interest: $\mathrm{E}\left[\hat{r}^{(p)}\right] = 0$ holds even for small samples and $\mathrm{Skew}\left[\hat{r}^{(p)}\right] = 0$ holds approximately. $\mathrm{Skew}\left[\hat{r}^{(p)}\right]$, the 3rd standardised moment, defines the degree of asymmetry of the distribution of $\hat{r}^{(p)}$. Therefore, $F_{\hat{\Theta}}(\Theta) = F_{\hat{r}^{(p)}}(0) = 0.5$ is true if n_b is large. The null hypothesis of

our test is that $\{s_{j,i}(m)\}_{i=1}^{n_b}$ is an independent sequence.

$$\Pr\left[|\hat{r}^{(p)}(P_k)| < |\hat{r}^{(p)}(P_1)|\right] = \frac{1}{2} \tag{A.4}$$

holds under the null hypothesis for a randomly chosen permutation P_k. For K randomly chosen permutations $P_{k_1}, \ldots, P_{k_K}$ we can derive

$$\Pr\left[\forall l(1 \leq s \leq K) : |\hat{r}^{(p)}(P_{k_s})| < |\hat{r}^{(p)}(P_1)|\right] = \frac{1}{2^K} \quad . \tag{A.5}$$

On basis of this equation a confidence interval can be established:

$$\Pr\left[-\Delta \leq \hat{r}^{(p)}(P_1) \leq \Delta\right] = 1 - \frac{1}{2^K} \tag{A.6}$$

with halfwidth

$$\Delta = \max_{1 \leq s \leq K} \left(|\hat{r}^{(p)}(P_{k_s})|\right) \quad . \tag{A.7}$$

If $\hat{r}^{(p)}(P_1)$ is not within the confidence interval, the null hypothesis must be rejected at confidence level $1 - \frac{1}{2^K}$.

The advantage of using this confidence interval is that the assumption of zero skewness is weaker than the assumption of a normal distribution. For only $K = 6$ permutations the confidence level is already > 0.95 and K can be regarded as a constant parameter. Therefore, the time complexity of the confidence interval calculation is the same as for the calculation of $\hat{r}^{(p)}(P_1)$ itself. For our purpose of estimating the overall batch size m for p independent replications this correlation test is performed on $\{s_{j,i}(m)\}_{i=1}^{\infty}$ for any j. Only if all test are positive m can be regarded as a valid batch size.

A.2 Models of Time Series

In Sections 4.4, 5.6 and 6.4 we apply time series to validate the performance of proposed estimation methods by comparing simulation results with exact values, which are derived in this section. Time series analysis, see e.g. [67-Ham94], is used to understand the underlying theory of a sequence of data. Models of time

series can have many forms. Two basic classes are the *autoregressive* (AR) and the *moving average* processes (MA). In both classes the *white noise* process Ψ_t is used to introduce randomness into the stochastic model. It consists of random variables with zero mean $\mathrm{E}[\Psi_t] = 0$, constant variance $\mathrm{Var}[\Psi_t] = \sigma_\Psi^2$ and is uncorrelated $\mathrm{Cov}[\Psi_{t_1}\Psi_{t_2}] = 0$, where $t_1 \neq t_2$. If Ψ_t is normally distributed with $F_{\Psi_t}(x) = \mathrm{N}(x; 0, \sigma_\Psi^2)$ it is called *Gaussian white noise* process.

An autoregressive process AR(p) is given by a weighted sum of p previous values, a constant value c and a new value taken of the white noise process:

$$X_t = c + \Psi_t + \sum_{i=1}^{p} \phi_i X_{t-i}, \tag{A.8}$$

where ϕ_i denotes the ith autoregressive parameter. An moving average process MA(q) is the weighted sum of q previous values of the white noise process, a constant value c and a new value taken of the white noise process:

$$X_t = c + \Psi_t + \sum_{i=1}^{q} \theta_i \Psi_{t-i} \tag{A.9}$$

where θ_i denotes the ith moving average parameter. Combining these two basic classes results in a broader class called *autoregressive moving average* (ARMA) processes. An ARMA(p, q) process is given by

$$X_t = c + \Psi_t + \sum_{i=1}^{q} \theta_i \Psi_{t-i} + \sum_{i=1}^{p} \phi_i X_{t-i}. \tag{A.10}$$

Conditions for stationarity and general formulas for the expected value, the variance and the autocorrelation function of the ARMA process are discussed in [19-BJR94]. Further generalisation are nonlinear ARMA models (NARMA) and autoregressive integrated moving average models (ARIMA) with its variations (e.g. seasonal ARIMA, fractional ARIMA).

Here we focus on special ARMA(k, k) processes with identical autoregressive and moving average parameters, which are defined by the geometrical series $\frac{1}{2^i}$,

which has easily determined properties. Thus, its definition is:

$$\Upsilon_t^{(k)} = 1 + \Psi_t + \sum_{i=1}^{k} \frac{1}{2^i}(\Upsilon_{t-i}^{(k)} + \Psi_{t-i}). \tag{A.11}$$

The order k is a positive integer and defines how many previous values are used to calculate a new value of this process. $\Upsilon_t^{(k)}$ is stationary for every value of k. Ψ_t is in this case a Gaussian white noise process with $\mathrm{Var}[\Psi_t] = 1$. We refer to this process as *geometrical* ARMA process.

A.2.1 Steady State Distribution of the First Order Process

In Section 6.4 we will compare quantile estimates with the exact CDF of the first order geometrical ARMA process, which is derived in this section. The first order geometrical ARMA process $\Upsilon_t^{(1)}$ is given by

$$\Upsilon_t^{(1)} = 1 + \Psi_t + \frac{1}{2}\Upsilon_{t-1}^{(1)} + \frac{1}{2}\Psi_{t-1}. \tag{A.12}$$

Because the process is stationary, its expected value and its variance is constant for every value of t. The expected value of $\Upsilon_t^{(1)}$ is

$$\mathrm{E}\left[\Upsilon_t^{(1)}\right] = 1 + \mathrm{E}[\Psi_t] + \frac{1}{2}\mathrm{E}\left[\Upsilon_{t-1}^{(1)}\right] + \frac{1}{2}\mathrm{E}[\Psi_{t-1}] \tag{A.13}$$

which leads to

$$\mathrm{E}\left[\Upsilon_t^{(1)}\right] = \frac{1}{1 - \frac{1}{2}} = 2 \tag{A.14}$$

by substituting $\mathrm{E}\left[\Upsilon_t^{(1)}\right] = \mathrm{E}\left[\Upsilon_{t-1}^{(1)}\right]$ and $\mathrm{E}[\Psi_t] = \mathrm{E}[\Psi_{t-1}] = 0$. The variance of $\Upsilon_t^{(1)}$ can be calculated in a similar way:

$$\begin{aligned} \mathrm{Var}\left[\Upsilon_t^{(1)}\right] &= \mathrm{E}\left[\left(1 + \Psi_t + \frac{1}{2}\Upsilon_{t-1}^{(1)} + \frac{1}{2}\Psi_{t-1} - \mathrm{E}\left[\Upsilon_t^{(1)}\right]\right)^2\right] \\ &= \frac{1}{4}\mathrm{Var}\left[\Upsilon_{t-1}^{(1)}\right] + \frac{1}{2}\mathrm{Cov}\left[\Upsilon_{t-1}^{(1)}, \Psi_{t-1}\right] \end{aligned} \tag{A.15}$$

The covariance $\mathrm{Cov}\left[\Upsilon_{t-1}^{(1)}, \Psi_{t-1}\right]$ is

$$\mathrm{Cov}\left[\Upsilon_{t-1}^{(1)}, \Psi_{t-1}\right] = \mathrm{E}\left[\left(\Upsilon_t^{(1)} - \mathrm{E}\left[\Upsilon_t^{(1)}\right]\right)(\Psi_t - \mathrm{E}[\Psi_t])\right] = \mathrm{Var}[\Psi_t] = 1 \tag{A.16}$$

and because $\mathrm{Var}\left[\Upsilon_t^{(1)}\right] = \mathrm{Var}\left[\Upsilon_{t-1}^{(1)}\right]$ and $\mathrm{Cov}\left[\Upsilon_t^{(1)}, \Psi_t\right] = \mathrm{Cov}\left[\Upsilon_{t-1}^{(1)}, \Psi_{t-1}\right]$ Equation (A.15) evolves to

$$\mathrm{Var}\left[\Upsilon_t^{(1)}\right] = \frac{2}{3}\mathrm{Cov}\left[\Upsilon_t^{(1)}, \Psi_t\right] + \frac{5}{3} = \frac{7}{3}. \tag{A.17}$$

Adding or multiplying a constant value to a normally distributed random variable or adding two independent normally distributed random variables results in a normally distributed random variable with different mean and/or variance. From [94-MF00]:

- $c + \mathrm{N}\left(x; \mu, \sigma^2\right) = \mathrm{N}\left(x; \mu + c, \sigma^2\right)$
- $c \cdot \mathrm{N}\left(x; \mu, \sigma^2\right) = \mathrm{N}\left(x; \mu, \sigma^2 \cdot c^2\right)$
- $\mathrm{N}\left(x; \mu_1, \sigma_1^2\right) + \mathrm{N}\left(x; \mu_2, \sigma_2^2\right) = \mathrm{N}\left(x; \mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2\right)$

In Equation (A.11) only constant values are added to or multiplied by the Gaussian white noise process Ψ_t. Therefore, the following theorem arises:

Theorem A.2.1 *The CDF of* $\Upsilon_t^{(1)}$ *is* $F_{\Upsilon_t^{(1)}}(x) = \mathrm{N}\left(x; 2, \frac{7}{3}\right)$.

Proof By repeatedly replacing $\Upsilon_t^{(1)}$ with its definition, it is possible to find a form of $\Upsilon_t^{(1)}$, which is based on a pattern described by two infinite series.

$$\begin{aligned}
\Upsilon_t^{(1)} &= 1 + \Psi_t + \frac{1}{2}\Psi_{t-1} + \frac{1}{2}\Upsilon_{t-1}^{(1)} \\
&= \frac{1}{1} + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots \\
&\quad + \Psi_t + \frac{1}{1}\Psi_{t-1} + \frac{1}{2}\Psi_{t-2} + \frac{1}{4}\Psi_{t-3} + \cdots \\
&= \sum_{j=0}^{\infty}\left(\frac{1}{2^j}\right) + \Psi_t + \sum_{j=0}^{\infty}\left(\frac{1}{2^j}\Psi_{t-j}\right)
\end{aligned} \tag{A.18}$$

The probability distribution function of Ψ_t is $F_{\Psi_t}(x) = \mathrm{N}(x; 0, 1)$. By using this

result, the probability distribution function of $\Upsilon_t^{(1)}$ can be determined.

$$\begin{aligned} F_{\Upsilon_t^{(1)}}(x) &= \sum_{j=0}^{\infty}\left(\frac{1}{2^j}\right) + \mathrm{N}(x;0,1) + \sum_{j=0}^{\infty}\left(\frac{1}{2^j}\mathrm{N}(x;0,1)\right) \\ &= \mathrm{N}\left(x; \sum_{j=0}^{\infty}\left(\frac{1}{2^j}\right), 1 + \sum_{j=0}^{\infty}\left(\frac{1}{4^j}\right)\right) \\ &= \mathrm{N}\left(x; \frac{1}{1-\frac{1}{2}}, 1 + \frac{1}{1-\frac{1}{4}}\right) \\ &= \mathrm{N}\left(x; 2, \frac{7}{3}\right) \end{aligned} \tag{A.19}$$

∎

We have proved that $F_{\Upsilon_t^{(1)}}(x) = \mathrm{N}\left(x; 2, \frac{7}{3}\right)$ is the expected steady state distribution of the first order geometrical ARMA process. This result can be used to verify estimations of $F_{\Upsilon_t^{(1)}}(x)$.

A.2.2 Steady State Distribution of the Second Order Process

In Section 6.4 we will compare quantile estimates with the exact CDF of the second order geometrical ARMA process, which is derived in this section. The second order geometrical ARMA process $\Upsilon_t^{(2)}$ is defined by

$$\Upsilon_t^{(2)} = 1 + \Psi_t + \frac{1}{2}\Upsilon_{t-1}^{(2)} + \frac{1}{4}\Upsilon_{t-2}^{(2)} + \frac{1}{2}\Psi_{t-1} + \frac{1}{4}\Psi_{t-2}. \tag{A.20}$$

As for the first order process, the expected value of $\Upsilon_t^{(2)}$ can be determined by

$$\mathrm{E}\left[\Upsilon_t^{(2)}\right] = 1 + \frac{1}{2}\mathrm{E}\left[\Upsilon_{t-1}^{(2)}\right] + \frac{1}{4}\mathrm{E}\left[\Upsilon_{t-2}^{(2)}\right] = \frac{1}{1-\frac{1}{2}-\frac{1}{4}} = 4. \tag{A.21}$$

The calculation of the variance is more difficult in the second order case than in the first order case.

$$\begin{aligned} \mathrm{Var}\left[\Upsilon_t^{(2)}\right] &= \mathrm{E}\left[\left(\Upsilon_t^{(2)} - \mathrm{E}\left[\Upsilon_t^{(2)}\right]\right)^2\right] \qquad \text{(A.22)} \\ &= 9 - \frac{9}{2}\mathrm{E}\left[\Upsilon_t^{(2)}\right] + \frac{9}{16}\mathrm{E}\left[\Upsilon_t^{(2)}\right]^2 + \frac{5}{16}\mathrm{Var}\left[\Upsilon_t^{(2)}\right] + \frac{21}{16}\mathrm{Var}\left[\Psi_t\right] \\ &\quad + \frac{1}{4}\mathrm{Cov}\left[\Upsilon_t^{(2)}, \Upsilon_{t-1}^{(2)}\right] + \frac{5}{8}\mathrm{Cov}\left[\Upsilon_t^{(2)}, \Psi_t\right] + \frac{1}{4}\mathrm{Cov}\left[\Upsilon_t^{(2)}, \Psi_{t-1}\right] \end{aligned}$$

This result shows, that more knowledge about the covariance of the process is needed. The unknown parts of this equation are

$$\begin{aligned}
\mathrm{Cov}\left[\Upsilon_t^{(2)}, \Psi_t\right] &= \mathrm{E}\left[\left(\Upsilon_t^{(2)} - \mathrm{E}\left[\Upsilon_t^{(2)}\right]\right)\left(\Psi_t - \mathrm{E}\left[\Psi_t\right]\right)\right] \\
&= \mathrm{Var}\left[\Psi_t\right] = 1; \qquad \text{(A.23)} \\
\mathrm{Cov}\left[\Upsilon_t^{(2)}, \Psi_{t-1}\right] &= \mathrm{E}\left[\left(\Upsilon_t^{(2)} - \mathrm{E}\left[\Upsilon_t^{(2)}\right]\right)\left(\Psi_{t-1} - \mathrm{E}\left[\Psi_{t-1}\right]\right)\right] \\
&= \frac{1}{2}\mathrm{Cov}\left[\Upsilon_t^{(2)}, \Psi_t\right] + \frac{1}{2}\mathrm{Var}\left[\Psi_t\right] = 1; \qquad \text{(A.24)} \\
\mathrm{Cov}\left[\Upsilon_t^{(2)}, \Upsilon_{t-1}^{(2)}\right] &= \mathrm{E}\left[\left(\Upsilon_t^{(2)} - \mathrm{E}\left[\Upsilon_t^{(2)}\right]\right)\left(\Upsilon_{t-1}^{(2)} - \mathrm{E}\left[\Upsilon_{t-1}^{(2)}\right]\right)\right] \\
&= \frac{1}{2}\mathrm{Var}\left[\Upsilon_t^{(2)}\right] + \frac{1}{4}\mathrm{Cov}\left[\Upsilon_t^{(2)}, \Upsilon_{t-1}^{(2)}\right] + \frac{3}{4} \\
&= \frac{2}{3}\mathrm{Var}\left[\Upsilon_t^{(2)}\right] + 1. \qquad \text{(A.25)}
\end{aligned}$$

Further more, $\mathrm{E}\left[\Upsilon_t^{(2)}\right]^2 = 16$ and $\mathrm{Var}\left[\Psi_t\right] = 1$. With the help of this results the variance can now be calculated:

$$\mathrm{Var}\left[\Upsilon_t^{(2)}\right] = \frac{39}{16} - \frac{23}{48}\mathrm{Var}\left[\Upsilon_t^{(2)}\right] = \frac{117}{25}. \qquad \text{(A.26)}$$

Because Ψ_t is a Gaussian white noise process, again, we can show that $\Upsilon_t^{(2)}$ is normally distributed.

Theorem A.2.2 *The CDF of* $\Upsilon_t^{(2)}$ *is* $F_{\Upsilon_t^{(2)}}(x) = \mathrm{N}\left(x; 4, \frac{117}{25}\right)$.

Proof $\Upsilon_t^{(2)}$ can be denoted on basis of infinite series.

$$\begin{aligned}
\Upsilon_t^{(2)} &= 1 + \Psi_t + \frac{1}{2}\Psi_{t-1} + \frac{1}{4}\Psi_{t-2} + \frac{1}{2}\Upsilon_{t-1}^{(2)} + \frac{1}{4}\Upsilon_{t-2}^{(2)} \\
&= \frac{1}{1} + \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{5}{16} + \frac{8}{32} + \frac{13}{64} + \cdots \\
&\quad + \Psi_t + \frac{1}{1}\Psi_{t-1} + \frac{2}{2}\Psi_{t-2} + \frac{3}{4}\Psi_{t-3} + \frac{5}{8}\Psi_{t-4} + \frac{8}{16}\Psi_{t-5} + \frac{13}{32}\Psi_{t-6} + \cdots \\
&= 2\sum_{i=0}^{\infty}\left(\frac{\mathrm{Fib}_i}{2^i}\right) + \Psi_t + \sum_{i=0}^{\infty}\left(\frac{\mathrm{Fib}_{i+2}}{2^i}\Psi_{t-i-1}\right) \qquad \text{(A.27)}
\end{aligned}$$

The probability distribution function of $\Upsilon_t^{(2)}$ is, therefore, given by

$$\begin{aligned} F_{\Upsilon_t^{(2)}}(x) &= 2\sum_{i=0}^{\infty}\left(\frac{\mathrm{Fib}_i}{2^i}\right) + \mathrm{N}(x;0,1) + \sum_{i=0}^{\infty}\left(\frac{\mathrm{Fib}_{i+2}}{2^i}\mathrm{N}(x;0,1)\right) \\ &= \mathrm{N}\left(x; 2\sum_{i=0}^{\infty}\left(\frac{\mathrm{Fib}_i}{2^i}\right), 1 + \sum_{i=0}^{\infty}\left(\frac{\mathrm{Fib}_{i+2}}{2^i}\right)^2\right) \end{aligned} \tag{A.28}$$

This equation contains two geometrical series with Fibonacci numbers in the numerator. The Fibonacci numbers are defined by $\mathrm{Fib}_i = \mathrm{Fib}_{i-1} + \mathrm{Fib}_{i-2}$, where $\mathrm{Fib}_0 = 0$ and $\mathrm{Fib}_1 = 1$. Binet's formula, see e.g [134-Vor02], is a closed form of the Fibonacci numbers:

$$\mathrm{Fib}_i = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^i - \left(\frac{1-\sqrt{5}}{2}\right)^i\right). \tag{A.29}$$

And therefore

$$\begin{aligned} \mathrm{Fib}_{i+2} &= \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^{i+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{i+2}\right) \\ &= \frac{3+\sqrt{5}}{2\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^i - \frac{3-\sqrt{5}}{2\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^i. \end{aligned} \tag{A.30}$$

By substituting Fib_i and Fib_{i+2} by the closed formula the two infinite series of

Equation (A.28) can be calculated.

$$\begin{aligned}\sum_{i=0}^{\infty}\left(\frac{\mathrm{Fib}_i}{2^i}\right) &= \frac{1}{\sqrt{5}}\sum_{i=0}^{\infty}\left(\frac{1+\sqrt{5}}{4}\right)^i - \frac{1}{\sqrt{5}}\sum_{i=0}^{\infty}\left(\frac{1-\sqrt{5}}{4}\right)^i \\ &= \frac{1}{\sqrt{5}}\cdot\frac{1}{1-\frac{1+\sqrt{5}}{4}} - \frac{1}{\sqrt{5}}\cdot\frac{1}{1-\frac{1-\sqrt{5}}{4}} \\ &= 2 \qquad \text{(A.31)}\end{aligned}$$

$$\begin{aligned}\sum_{i=0}^{\infty}\left(\frac{\mathrm{Fib}_{i+2}}{2^i}\right)^2 &= \frac{7+3\sqrt{5}}{10}\sum_{i=0}^{\infty}\left(\frac{3+\sqrt{5}}{8}\right)^i \\ &\quad + \frac{7-3\sqrt{5}}{10}\sum_{i=0}^{\infty}\left(\frac{3-\sqrt{5}}{8}\right)^i \\ &\quad - \frac{2}{5}\sum_{i=0}^{\infty}\left(-\frac{1}{4}\right)^i \\ &= \frac{7+3\sqrt{5}}{10}\cdot\frac{8}{5-\sqrt{5}} + \frac{7-3\sqrt{5}}{10}\cdot\frac{8}{5+\sqrt{5}} - \frac{8}{25} \\ &= \frac{92}{25} \qquad \text{(A.32)}\end{aligned}$$

And finally, the probability distribution function of $\Upsilon_t^{(2)}$ is

$$F_{\Upsilon_t^{(2)}}(x) = \mathrm{N}\left(x; 2\cdot 2, 1+\frac{92}{25}\right) = \mathrm{N}\left(x; 4, \frac{117}{25}\right) \qquad \text{(A.33)}$$

∎

We have proved that $F_{\Upsilon_t^{(2)}}(x) = \mathrm{N}\left(x; 4, \frac{117}{25}\right)$ is the expected steady state distribution of the second order geometrical ARMA process. This result can be used to verify estimations of $F_{\Upsilon_t^{(2)}}(x)$.

A.3 M/M/1 Queue

The M/M/1 queue is probably the most commonly used single server example of queueing systems. We follow Kendall's notation. Interarrival and service times are exponentially distributed, where λ denotes the mean arrival rate and μ denotes the mean service rate. The traffic intensity is $\rho = \frac{\lambda}{\mu}$. If $\rho < 1$ the queue is stable

Listing A.1: Pseudocode for the calculation of $p_{i,n}$.

```
0  float λ;
   float μ;
   int k;
   int n_max;
   // ***** λ, μ, k and n_max are user input. *****
5
   float ρ := λ/μ;
   float a := ρ/(ρ+1);
   float b := 1−a;

10 if (k ≤ 0) k := 1;
   for (int n := 1; n ≤ n_max; ++n)
      for (int i := n_max; i ≥ 1; −−i){
         if (n ≤ k){
            if (i = n) p_{i,n} := 1; // rule (1)
15          else p_{i,n} := 0;
         }else{
            if (i > n) p_{i,n} := 0;
            if (i = n) p_{i,n} := a^{n−k}; // rule (2)
            if ((i < n) ∧ (i ≠ 1)) p_{i,n} := a · p_{i−1,n−1} + b · p_{i+1,n}; // rule (3)
20          if (i = 1) p_{i,n} := b/a · p_{i+1,n}; // rule (4)
         }
      }
```

and steady state measures can be calculated. The mean number of jobs in system is $\frac{\rho}{1-\rho}$. The mean response time (waiting + service time) is $\mathrm{E}[R_\infty] = \frac{1}{\mu(1-\rho)}$. The CDF of the response time in steady state is $F_{R_\infty}(x) = 1 - e^{-x/\mathrm{E}[R_\infty]}$. These steady state properties are well known, see for example in [75-Jai91].

In [79-KL85] the transient behaviour of the M/M/1 queue is discussed. An numerical approach is described that calculates $P_k(n', i)$, the probability of i customers present in the system at the arrival of the n'th (non initial) customer with k customers already present at time 0, relying on the fact that underlying random variables behave like a Markov chain. We are also interested in performance measures regarding the initial customers and the system's response time. Thus, we use a slightly modified numerical approach, which is described in Listing A.1.

The user has to set all the parameters up to Line 4. Then, $a = \Pr[A < S] = \frac{\rho}{\rho+1}$ and $b = \Pr[S < A] = 1 - a$ can be calculated, where A and S represent

interarrival and service times. The output will be given in a matrix of $p_{i,n}$, where $1 \leq n \leq n_{\max}$ and $1 \leq i \leq n_{\max}$. Note that n includes both, initial and regular customers. $p_{i,n}$ is the probability that customer n (initial or not) waited for i customers to finish service, including himself, at the end of his own service. In Line 10 the value of k is adjusted for the case that no customers are present at $t = 0$, i.e. $k \leq 0$. The first of all customers, initial or not, never has to wait for other customers. Thus, the cases $k = 0$ and $k = 1$ do not have to be processed differently. The actual arrival time of the first customer does not influence the calculation of $p_{i,n}$. The outer loop in Line 11 increases n. The inner loop in Line 12 decreases i. This order is efficient for the calculation of $p_{i,n}$. The statement in Line 13 separates the initial and the non initial customers. For the initial customers $p_{i,n}$ is one or zero. This is because no initial customer can finish its service before all initial customers are present. Line 17 to Line 20 describe $p_{i,n}$ for non initial customers. Line 18 covers the case where no customer has left the system and here $p_{n,n} = a^{n-k}$. The general case is given in Line 19. Here, $p_{i,n}$ depends on $p_{i-1,n-1}$ and $p_{i+1,n}$. In Line 20 the case of an empty system is regarded and $p_{i,n}$ depends only on $p_{i+1,n}$. More details and further explanations of the rules can be found in [79-KL85]. Note that the complexity of Listing A.1 is much reduced compared with the algorithm in [79-KL85] and it covers both, $k = 0$ and $k > 0$. This is achieved by avoiding special cases by a simpler algorithmic approach.

In Figure A.1 we can see an example of the matrix of all $p_{i,n}$, where $\lambda = 0.5$, $\mu = 1$, $k = 2$ and $n_{\max} = 6$ so that $a = 0.\bar{3}$ and $b = 0.\bar{6}$. In the table n is given on the horizontal scale and i is given on the vertical scale. Rounded values of $p_{i,n}$ are given in the top left corner of each entry of the table. In the bottom right corner the applied rule is given, as stated in lines 14, 18, 19 and 20 in Listing A.1. For $n = 6$ and $i = 3$ the arrows are an example of how Line 19 is calculated. And for $n = 3$ and $i = 1$ the arrow is an example of how Line 20 is calculated.

If all $p_{i,n}$ are known the CDF of the system's response time R_n for the nth

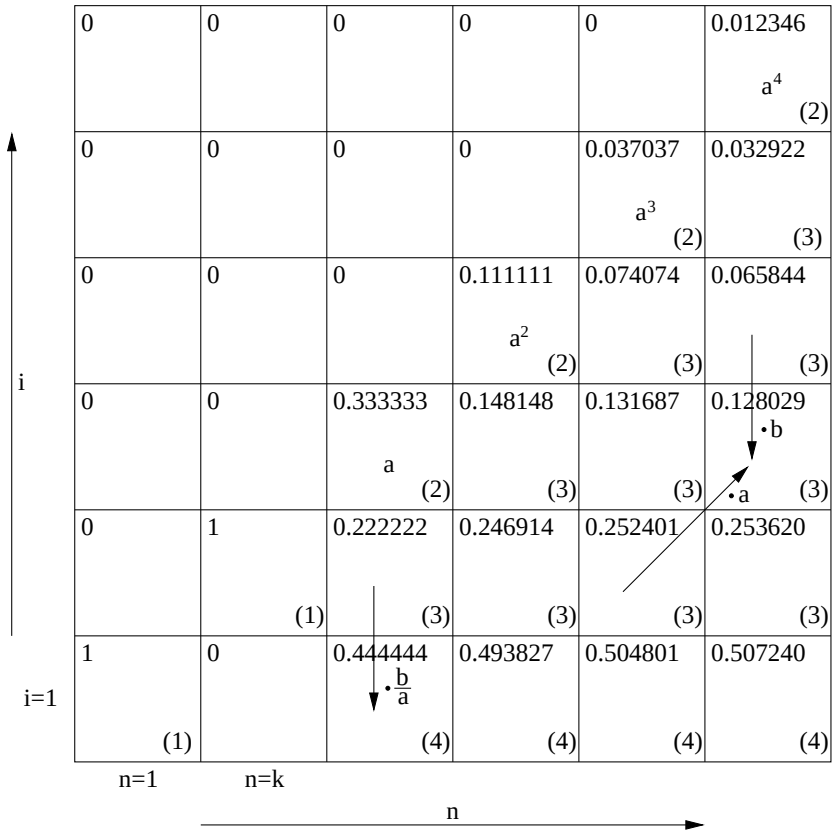


Figure A.1: $p_{i,n}$ calculated by Listing A.1, where $\lambda = 0.5$, $\mu = 1$, $k = 2$ and $n_{\text{max}} = 6$ so that $a = 0.\bar{3}$ and $b = 0.\bar{6}$.

customer leaving the system can be calculated. Every customer receives an exponentially distributed service time. Thus, the response time distribution can be found by summing i exponential random variables:

$$F_{R_n}(x) = \sum_{i=1}^{n} \left(p_{i,n} \sum_{1}^{i} \text{Exp}\left(x;\mu\right) \right) = \sum_{i=1}^{n} \left(p_{i,n} \text{Erlang}\left(x;\mu,i\right) \right). \tag{A.34}$$

Again, n includes initial and regular customers. The inner sum of exponential distributions adds up the service times of the i customers in the queue. This overall waiting time is Erlang distributed and has to be weighted by the probability of i customers in the queue. The outer sum adds up all weighted waiting times. This formula can be calculated for any x and it can be used to validate estimates of $F_{R_n}(x)$.

Listing A.2: Pseudocode for approximation of $F_{R_\infty}(x)$ in an M/E$_2$/1 queue.

```
long double ME2 1_response(long double X, long double λ, long double μ) const{
  // Note: mean service time is 2/μ
  if ((X < 0) ∨ (λ ≤ 0) ∨ (μ ≤ 0)) throw error("Invalid Parameter.");
  long double ρ := 2*λ/μ;
  if (ρ ≥ 1) throw error("Unstable Model.");

  long double part1:=cosh((X*sqrt(λ*(4*μ+λ)))/2)*exp(((-2*μ+λ)*X)/2);
  long double part2:=(-2*μ+λ)*sinh((X*sqrt(λ*(4*μ+λ)))/2)*exp(((-2*μ+λ)*X)/2);
  long double part3:=sqrt(λ*(4*μ+λ));

  return 1-part1+(part2/part3);
}
```

A.4 M/E$_2$/1 Queue

In many practical situations an exponential assumption of the service time may be rather limiting. The M/E$_k$/1 queue has an Erlang type k distribution. For $k \to \infty$ the service time's distribution is deterministic and for $k = 1$ it is exponential. Using $k = 2$ the coefficient of variation of the service time is $\frac{1}{\sqrt{2}}$, the mean inter-arrival time is $\frac{1}{\lambda}$. The service time is $\frac{2}{\mu}$ because every customer has a single service consisting of the sum of two independent and identically distributed exponential random variables.

To calculate the CDF $F_{R_\infty}(x)$ of the steady state response time the general approach for the M/G/1 queue can be applied, for details see e.g. [66-GH98]. In the general case the relationship between the Laplace-Stieltjes transform of the service and response times, $G(s)$ and $W(s)$, is given by Pollaczek-Khintchine transform equation

$$W(s) = \frac{(1-\rho)sG(s)}{s - \lambda(1 - G(s))}. \tag{A.35}$$

In the case of the M/E$_2$/1 queue the Laplace-Stieltjes transform of the service time distribution is

$$G(s) = \left(\frac{\mu}{\mu + s}\right)^2. \tag{A.36}$$

Then, the Pollaczek-Khintchine transform for the response time simplifies to

$$W(s) = \frac{(\mu - 2\lambda)\mu}{s(\mu^2 + 2\mu s + s^2 - 2\lambda\mu - \lambda s)}. \tag{A.37}$$

Listing A.3: Pseudocode for approximation of $F^{-1}_{R_\infty}(y)$ in an M/E$_2$/1 queue.

```
long double inv_ME21_response(long double y, long double λ, long double μ) const{
  if ((0 > y) ∨ (1 ≤ y) ∨ (λ ≤ 0) ∨ (μ ≤ 0)) throw error("Invalid Parameter.");
  long double ρ := 2*λ/μ;
  if (ρ ≥ 1) throw error("Unstable Model.");

  if (y = 0) return 0; // special case
  const long double allowed_difference:=0.0000000001;

  // lower bound
  long double l:=0;
  long double F_l:=0;

  // upper bound
  long double u:=1;
  long double F_u:=ME21_response(u,λ,μ);
  while (F_u ≤ y) {u:=u*2; F_u:=ME21_response(u,λ,μ);} // find u with F_R∞(u) > y

  // binary search based on ME21_response
  long double c:=l+(u-l)/2; // centre of l and u
  long double F_c:=ME21_response(c,λ,μ);
  while (u-l>allowed_difference){
    if (F_c > y) {u:=c; F_u:=F_c;}
    else         {l:=c; F_l:=F_c;}
    c:=l+(u-l)/2; F_c:=ME21_response(c,λ,μ);
  }
  return c;
}
```

By inverting the Laplace-Stieltjes transform using Maple we obtain

$$F_{R_\infty}(x) = 1 - \cosh\left(x\frac{a}{2}\right)e^{xb} + \frac{\lambda - 2\mu}{a}\sinh\left(x\frac{a}{2}\right)e^{xb}, \tag{A.38}$$

where $a = \sqrt{\lambda(4\mu + \lambda)}$ and $b = \frac{(\lambda - 2\mu)}{2}$. For calculations we used the software tool Maple and for further explanations see e.g. [66-GH98]. To compare simulation results with the expected distribution a method is needed that calculates Equation (A.38). Pseudocode of an implementation of this method is given in Listing A.2.

To calculate coverage of interval estimates of quantiles the inverse distribution $F^{-1}_{R_\infty}(y)$ is needed. It could be calculated by inverting Equation (A.38). However, this leads to a complex formula which is difficult to implement. We found that a binary search is practical. Let $F_{R_\infty}(l) \leq F_{R_\infty}(x) \leq F_{R_\infty}(u)$, where l and u are upper and lower bounds and $x = F^{-1}_{R_\infty}(y)$ is the value of interest. l and u are increased and decreased until their distance is smaller than a predefined threshold.

Then, $l+(u-l)/2 \approx x$ can be assumed. Pseudocode of this binary search is given in Listing A.3. For our purpose of coverage analysis the run time of this method is acceptable.

The transient behaviour of the M/E_k/1 queue can be calculated with a similar approach to Listing A.1. Here, a matrix of the size $p_{i,n}$ has to be extended to $1 \leq i \leq kn_{\text{max}}$ and $1 \leq n \leq n_{\text{max}}$ to incorporate the k exponential services of the Erlang distribution. Details of the calculation of $p_{i,n}$ are given in [78-Kel85] and [92-McN91]. The calculation of $F_{R_n}(x)$ can be done by the extension of Equation (A.34) to allow for the new size of the matrix $p_{i,n}$.

A.5 M/H_2/1 Queue

The service time of an M/H_k/1 queue is governed by a hyperexponential distribution of dimension k. We focus on $k = 2$ and a mean interarrival time of $\lambda = 1$. Let p denote the probability of choosing a service rate μ_1 and $1-p$ the probability of choosing a service rate μ_2 of exponential distributions. Then, the overall mean service time m is given by

$$m = \frac{p}{\mu_1} + \frac{1-p}{\mu_2} \tag{A.39}$$

and the variance is

$$\sigma^2 = 2\left(\frac{p}{\mu_1^2} + \frac{1-p}{\mu_2^2}\right) - \left(\frac{p}{\mu_1} + \frac{1-p}{\mu_2}\right)^2. \tag{A.40}$$

To fully specify the parameters p, μ_1 and μ_2 three conditions are needed. For the first condition we set the wanted value of m. The second condition is a squared

m	p	μ_1	μ_2
0.5	0.2113248654	0.8452994616	3.154700538
0.75	0.2113248654	0.5635329745	2.103133692
0.9	0.2113248654	0.4696108120	1.752611410

Table A.1: Parameters of the M/H_2/1 queue.

Listing A.4: Pseudocode for approximation of $F_{R_\infty}(x)$ in an M/H$_2$/1 queue.

```
long double MH2 1_response(long double X, long double λ, long double μ1,
                           long double μ2, long double p) const{
  if ((0 > X) ∨ (λ ≤ 0) ∨ (μ1 ≤ 0) ∨ (μ2 ≤ 0) ∨ (0 > p) ∨ (p > 1))
    throw error("Invalid Parameter.");

  long double q := 1−p;
  long double mean := (p/μ1)+(q/μ2);
  long double var := (2∗p)/μ1^2+(2∗(1−p))/μ2^2−((p/μ1)+(q/μ2))^2;

  long double ρ := λ ∗ mean;
  if (ρ ≥ 1) throw error("Unstable Model.");

  long double part1 := sqrt(μ1^2+(2∗λ∗μ1)−(2∗μ1∗μ2)+λ^2−(2∗λ∗μ2)+μ2^2
                       +(4∗λ∗p∗μ2)−(4∗λ∗μ1∗p));
  long double part2 := exp(((λ−μ1−μ2)∗X)/2);
  long double part3 := cosh(X∗part1/2);
  long double part4 := sinh(X∗part1/2);
  long double part5 := ((−2∗λ∗p∗μ2^2)−(μ1^2∗μ2)−(μ2∗λ∗μ1)+(μ1∗μ2^2)
                       −(4∗p^2∗μ2∗λ∗μ1)+(4∗μ2∗λ∗μ1∗p)
                       −(2∗p∗μ2^2∗μ1)+(2∗p^2∗μ2^2∗λ)+(2∗μ1^2∗p∗μ2)−(2∗μ1^2∗p∗λ)
                       +(2∗μ1^2∗p^2∗λ))/(part1∗μ1∗μ2);
  return 1−part2∗part3+part4∗part2∗part5;
}
```

coefficient of variance being 2. This is given by setting $\sigma^2 = 2m^2$. A standard assumption for the third condition is that of balanced means, which is given if $\frac{p}{\mu_1} = \frac{1-p}{\mu_2}$. For $m = \{0.5, 0.75, 0.9\}$ we receive the settings which are shown in Table A.1. They are computed by the software tool Maple.

As in the example of the M/E$_2$/1 queue, we calculate the CDF $F_{R_\infty}(x)$ of the steady state response time with the software tool Maple. Here, the Laplace-Stieltjes transform of the service time distribution is

$$G(s) = \frac{p\mu_1}{\mu_1 + s} + \frac{(1-p)\mu_2}{\mu_2 + s}. \tag{A.41}$$

Thus, the Pollaczek-Khintchine transform is

$$W(s) = \frac{\left(1 - \lambda\left(\frac{p}{\mu_1} + \frac{1-p}{\mu_2}\right)\right)\left(\frac{p\mu_1}{\mu_1+s} + \frac{(1-p)\mu_2}{\mu_2+s}\right)}{s - \lambda\left(1 - \frac{p\mu_1}{\mu_1+s} - \frac{(1-p)\mu_2}{\mu_2+s}\right)}. \tag{A.42}$$

By inverting the Laplace-Stieltjes transform we obtain

$$F_{R_\infty}(x) = 1 - \cosh\left(x\frac{a}{2}\right)e^{xc} + \frac{b}{a\mu_1\mu_2}\sinh\left(x\frac{a}{2}\right)e^{xc}, \tag{A.43}$$

Listing A.5: Pseudocode for approximation of $F_{R_\infty}^{-1}(y)$ in an M/H_2/1 queue.

```
long double inv_MH21_response(long double y, long double λ, long double μ1,
                              long double μ2, long double p) const{
  if ((0 > y) ∨ (1 ≤ y) ∨ (λ ≤ 0) ∨ (μ1 ≤ 0) ∨ (μ2 ≤ 0) ∨ (0 > p) ∨ (p > 1))
    throw error("Invalid Parameter.");

  long double q := 1−p;
  long double mean := (p/μ1)+(q/μ2);
  long double var := (2∗p)/μ1^2+(2∗(1−p))/μ2^2−((p/μ1)+(q/μ2))^2;

  long double ρ := λ ∗ mean;
  if (ρ ≥ 1) throw error("Unstable Model.");

  if (y = 0) return 0; // special case
  const long double allowed_difference:=0.0000000001;

  // lower bound
  long double l:=0;
  long double Fl:=0;

  // upper bound
  long double u:=1;
  long double Fu:=MH21_response(u,λ,μ1,μ2,p);
  while (Fu ≤ y) {u:=u∗2; Fu:=MH21_response(u,λ,μ1,μ2,p);} //find u with F_R∞(u) > y

  // binary search based on MH21_response.
  long double c:=l+(u−l)/2; // centre of l and u
  long double Fc:=MH21_response(c,λ,μ1,μ2,p);
  while (u−l>allowed_difference){
    if (Fc > y) {u:=c; Fu:=Fc;}
    else        {l:=c; Fl:=Fc;}
    c:=l+(u−l)/2; Fc:=MH21_response(c,λ,μ1,μ2,p);
  }
  return c;
}
```

where

$$\begin{aligned} a &= \sqrt{\mu_1^2 + 2\lambda\mu_1 - 2\mu_1\mu_2 + \lambda^2 - 2\lambda\mu_2 + \mu_2^2 + 4\lambda p\mu_2 - 4\lambda\mu_1 p}, \\ b &= -2\lambda p\mu_2^2 - \mu_1^2\mu_2 - \mu_2\lambda\mu_1 + \mu_1\mu_2^2 - 4p^2\mu_2\lambda\mu_1 + 4\mu_2\lambda\mu_1 p \\ &\quad -2p\mu_2^2\mu_1 + 2p^2\mu_2^2\lambda + 2\mu_1^2 p\mu_2 - 2\mu_1^2 p\lambda + 2\mu_1^2 p^2\lambda \quad \text{and} \\ c &= \frac{\lambda - \mu_1 - \mu_2}{2}. \end{aligned} \tag{A.44}$$

For further explanations see e.g. [66-GH98]. Pseudocode of an implementation of the calculation of $F_{R_\infty}(x)$ for any x is given in Listing A.4. This implementation is based on Equation (A.43).

For coverage analysis of the interval estimate of quantiles a method is needed

that calculates $F^{-1}_{R_\infty}(y)$ for any given y. As in the example of the M/E$_2$/1 queue we use a binary search based on the calculation of $F_{R_\infty}(x)$. Pseudocode of this method is given in Listing A.5.

A.6 Empirical Analysis of the Power of the Tests in Section 5.3

In Section 5.3 we discussed homogeneity tests and their application in truncation point detection methods of Chapter 5. The Anderson-Darling test was discussed and tested to give good performance for our purpose. Here, we would like to estimate the power of our application of the Anderson-Darling test and its use in a multiple comparisons approach.

The null hypothesis H_0 of all our Anderson-Darling 2-sample tests is that the two random samples of X_1 and X_2 are identically distributed. The associated alternative hypothesis is that the two random samples of X_1 and X_2 are differently distributed. There are four different situations when performing a homogeneity test, where α is the significance level and $1-\beta$ is the power of the test.

H_0 **is true but rejected:** This is a Type I error, its probability is α.

H_0 **is true and not rejected:** This is a correct result, its probability is $1-\alpha$.

H_0 **is false and rejected:** This is a correct result, its probability is $1-\beta$.

H_0 **is false but not rejected:** This is a Type II error, its probability is β.

The Anderson-Darling test, which is described in Section 5.3.2 is nonparametric. Here, we are interested in its performance when it is applied to common situations. Therefore, we determine α and β of our implementation of the Anderson-Darling 2-sample test empirically. For this purpose a series of experiments was done with random samples from uniform, normal and exponential distributions. The experiments are done for different sample size $p = \{30, 100, 200\}$. These

settings are selected because $p = 30$ is recommended to be the minimum number of replications when applying the methods of Chapter 5 and p selected between 100 and 200 is a good setting, see Section 5.5.2. For every setting 10^5 independent homogeneity tests are done. In Table A.2 empirical values of α are reported, which are determined by counting the number of rejections of a true H_0 in 10^5 independent Anderson-Darling 2-sample tests. All reported values are smaller than the chosen significance level $\alpha = 0.05$, thus, the probability of a Type I error is small and our implementation returns valid results if H_0 is true.

In a second series of experiments we determine the power of our implementation of the Anderson-Darling 2-sample test empirically. Again, $\mathrm{U}(x; 0, 1)$, $\mathrm{N}(x; 0, 1)$ and $\mathrm{Exp}(x; 1)$ are used. To force H_0 to be false, the distribution of the second random sample in each Anderson-Darling 2-sample test is displaced by $\Delta = \mathrm{E}[X_2] - \mathrm{E}[X_1]$. The values of Δ are increased, starting with Δ close to 0. Results are depicted in Figure A.2. Empirical values of the power $1 - \beta$ are determined by counting rejections of a false H_0 in 10^3 independent Anderson-Darling 2-sample tests for every Δ separately. The values of the empirical power are rising from a low level close to 0 towards 1. This demonstrates that the probability of a Type II error is shrinking with growing displacement in distribution, as we would expect.

So far we assessed the performance of a single Anderson-Darling 2-sample test. However, the algorithms of Chapter 5 involve multiple comparisons as well. The probability of not rejecting a true H_0 in one test is $1 - \alpha$. So, the probability of not rejecting a true H_0 in k independent tests is $(1 - \alpha)^k$. Thus, the

	$p = 30$	$p = 100$	$p = 200$
$\mathrm{U}(x; 0, 1)$	0.0077	0.00682	0.00672
$\mathrm{N}(x; 0, 1)$	0.01559	0.01339	0.01301
$\mathrm{Exp}(x; 1)$	0.02374	0.02028	0.01934

Table A.2: Empirical values of α determined by counting rejections of a true H_0 in 10^5 independent Anderson-Darling 2-sample tests.

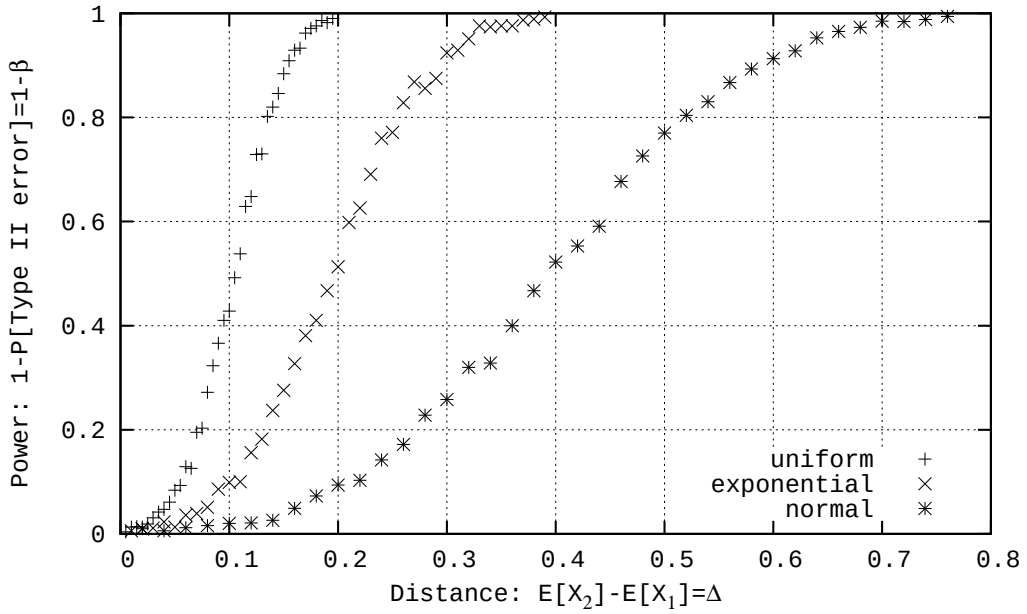
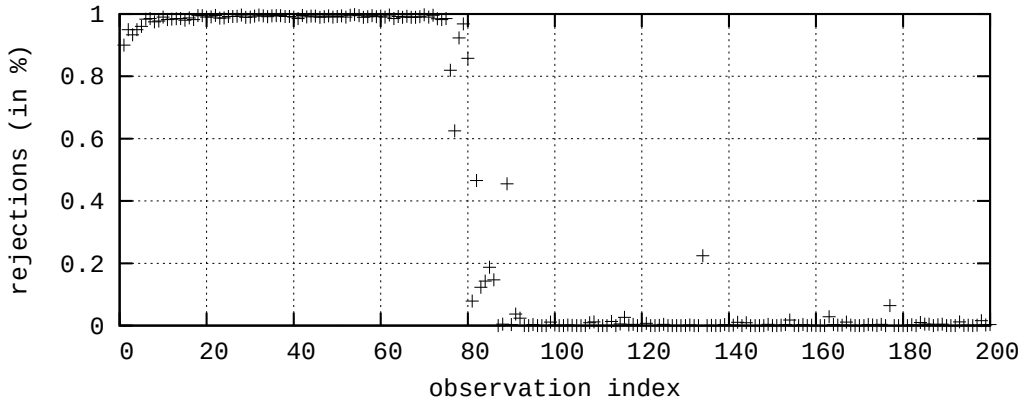
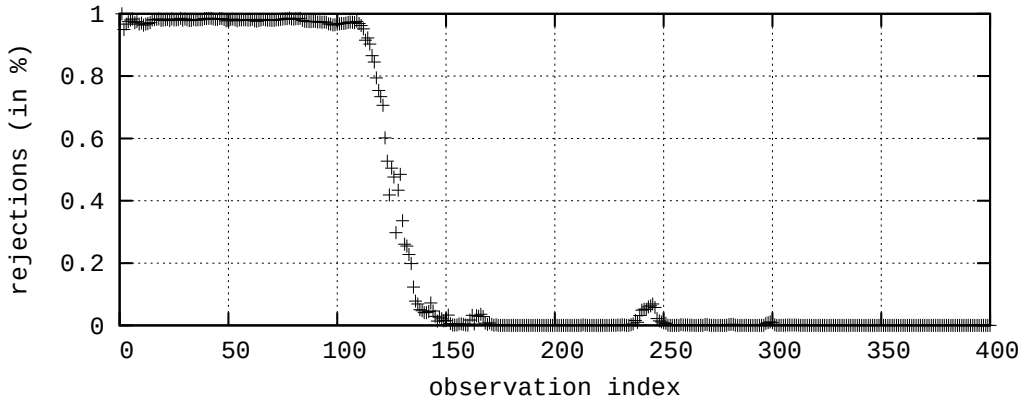


Figure A.2: Empirical values of the power $1-\beta$ determined by counting rejections of a false H_0 in 10^3 independent Anderson-Darling 2-sample tests.

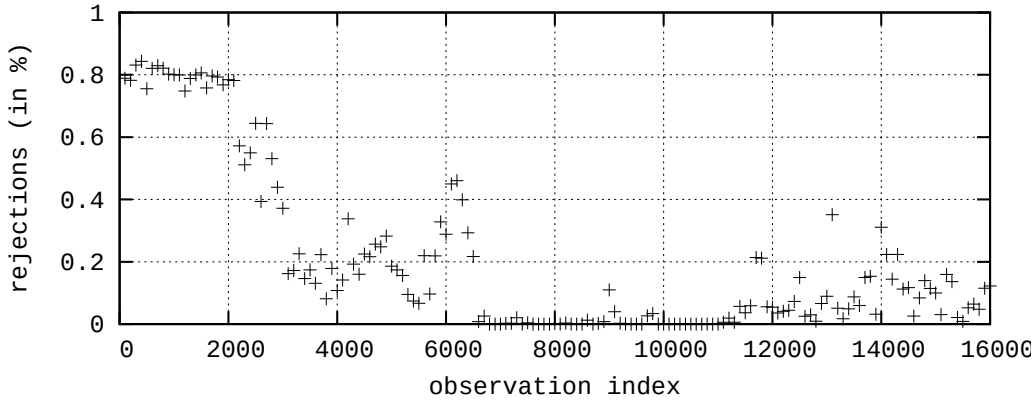
probability of rejecting a true H_0 in k independent tests is $\alpha_m = 1 - (1 - \alpha)^k$. α_m can be regarded as the experiment-wide significance level. These calculations assume independent homogeneity tests and, therefore, this result is likely to be a lower bound for our application. To assess α_m empirically for the multiple Anderson-Darling 2-sample tests we perform the algorithm of Listing 5.1. For every step of the algorithm we depicted the percentage of rejections of H_0 over time in Figure A.3. Note, that for this purpose all $i \cdot r$ Anderson-Darling 2-sample tests are performed in the ith step of the algorithm, unlike Line 16 of Listing 5.1. Each cross in Figure A.3 shows the rejections of H_0 based on just one simulation experiment, so they are not averaged results. To assess α_m empirically we have to make sure that H_0 is true, which means that the analysed process is in steady state. Thus, we are looking for the first observation index with no rejections of H_0 and assume that this is the beginning of the steady state.



(a) quadratic displacement: Equation (5.32)



(b) ARMA$(5, 5)$: Equation (5.33)



(c) bounded random walk: Equation (4.14)

Figure A.3: Percentage of rejections of H_0 for each step of the algorithm of Listing 5.1.

In the case of the quadratic displacement, see Equation (5.32) and Figure A.3(a), the first observation index with no rejections is at 87 in this simulation run. Until observation index 200 there are another 57 observation indexes where H_0 is not rejected by any of the Anderson-Darling 2-sample tests. So, we can derive that in this example the empirical α_m is $1 - (57/(200 - 87)) \approx 0.5$. In the case of the ARMA$(5, 5)$ process, see Equation (5.33) and Figure A.3(b), the first observation index with no rejections of H_0 is at 174. Until observation index 400 there are another 165 observation indexes where H_0 is not rejected by any of the Anderson-Darling 2-sample tests. So, we can derive that in this example the empirical α_m is $1 - (165/(400 - 174)) \approx 0.27$. In our last example we use a bounded random walk, see Equation (4.14) and Figure A.3(c). Because of high computational effort we depict only observation indexes every 100 steps and include those in our analysis. The first observation index with no rejections of H_0 is at 6800. Until observation index 16000 there are another 11 observation indexes (with step size 100!) where H_0 is not rejected by any of the Anderson-Darling 2-sample tests. In this example the empirical α_m is $1 - (11/(160 - 68)) \approx 0.88$. As we can see, the empirical value of α_m depends mainly on three factors, (a) the number of Anderson-Darling 2-sample tests given by $i \cdot r$, (b) the autocorrelation structure of the underlying process and (c) the fact that one random sample is used in all of the Anderson-Darling 2-sample tests. The factors (b) and (c) introduce high correlation into the results of the Anderson-Darling 2-sample tests of one step of the algorithm.

The empirical values of α_m are quite large. In consequence the probability of a Type I error, rejecting a true H_0, is large. However, this may well be acceptable when trying to find a truncation point. In practise it is not a problem if H_0 is falsely rejected for a number of possible truncation points, because we need to find only one valid candidate and usually it is better to be deeper in steady state. Therefore, we use the criterion that none of the Anderson-Darling 2-sample tests should reject

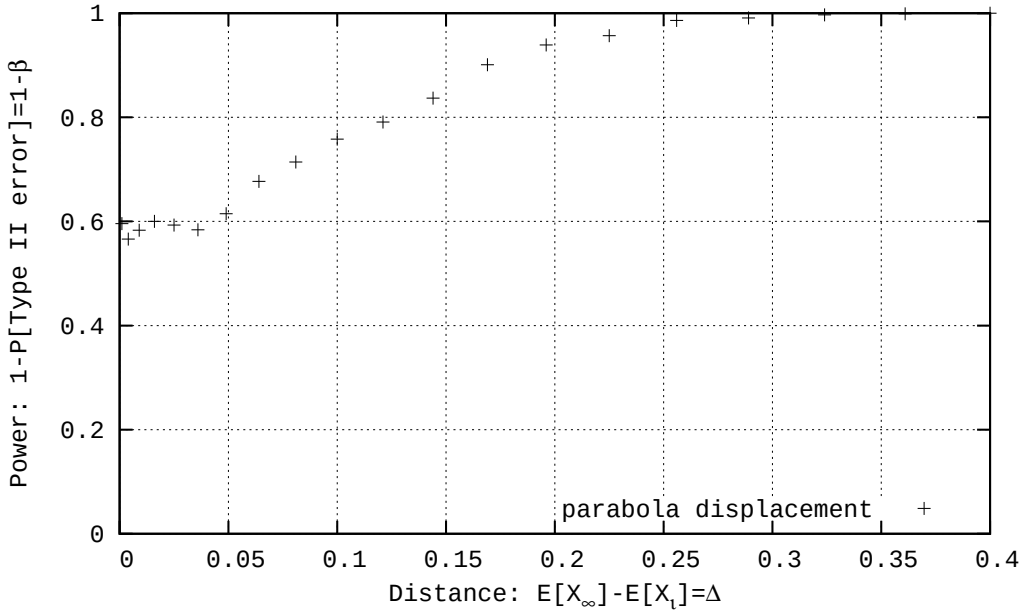
H_0 to indicate that steady state is reached. This is the most conservative approach and it avoids the need for defining α_m.

This most conservative approach leads to good results in all our examples of Section 5.6 and might be a good choice for many other simulation models. Even in the example of the bounded random walk, where truncation points are extremely large, it showed convincing results. Note, that in this example at observation index $i = 6800$ and $r = 10$ a number of 68000 Anderson-Darling 2-sample tests did not reject H_0. This shows again, that the performed tests at one step of the algorithm are extremely dependent on each other. This is the reason why the more advanced algorithms in Listing 5.2 and Listing 5.3 avoid doing $i \cdot r$ homogeneity tests in the ith step of the algorithm. These algorithms perform a constant number of just $r+1$ (resp. r) Anderson-Darling 2-sample tests in each step leading to better time complexity and storage requirements. The remaining Anderson-Darling 2-sample tests are done with random samples of carefully selected observation indexes.

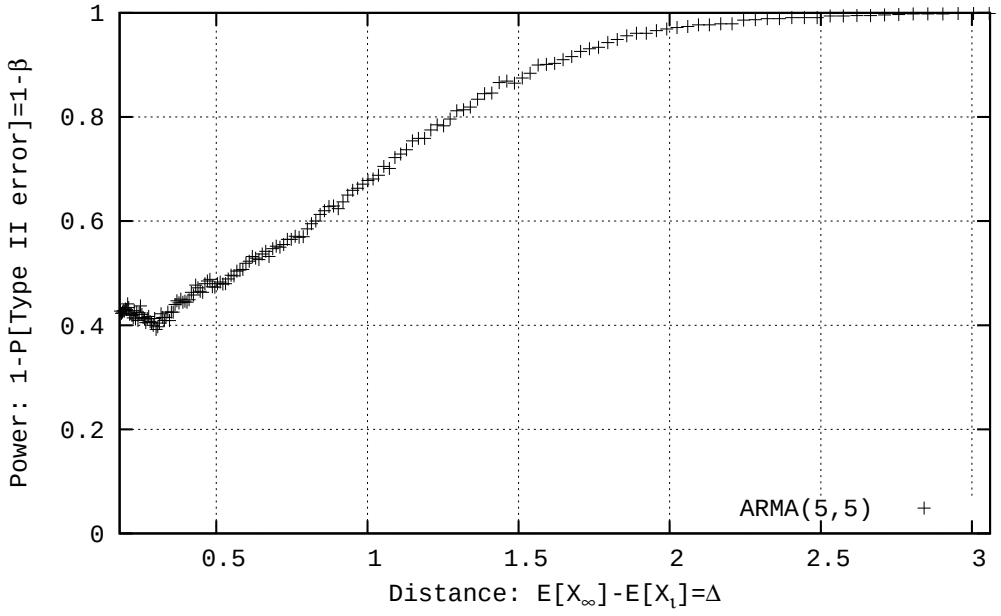
A serious problem arises, if the empirical value of α_m tends to 1. In this situation a true H_0 would always be rejected and the algorithm would not be able to detect steady state. A behaviour like this was not detected in any of the experiments performed for this research work. However, in Figure A.3(c) we saw that the number of rejections is rising again after observation index 6800, the smallest index without rejections, was reached. This indicates that, for an unnecessarily large r, the problem of the empirical α_m tending to 1 could arise. In this situation the graph of the percentage of rejections drawn over the observation indexes will still have a minimum. We recommend choosing the truncation point at this minimum, the observation index where the least rejections are done. Regarding the experiments for this research work a situation like this seems to be improbable. But still, it is possible for the algorithm of Listing 5.1, because the number of homogeneity tests is given by $i \cdot r$ and, therefore, increasing in each step. A situation where α_m tends to 1 is practically impossible for the algorithms of Listing 5.2 and

Listing 5.3, because in these cases the number of homogeneity tests is constant for every step. However, to guarantee that the algorithms stop in any case an alternative stopping criterion is introduced that simply stops the tests if the simulation horizon is larger than one would expect (see Listing 5.1 in Line 11, Listing 5.2 in Line 13 and Listing 5.3 in Line 13).

Assessing α_m implies that H_0 is true and it can be done only in steady state. The counterpart $1 - \beta_m$, the experiment-wide power, implies a false H_0 and it can be assessed during the transient phase. In Figure A.2 we saw that the power $1 - \beta$ is dependent on the displacement of the distribution. We have to take this into account when determining $1 - \beta_m$ empirically. As examples we chose a quadratic displacement (see Equation (5.32)) and an ARMA$(5, 5)$ process (see Equation (5.33)), both governed by a transient mean value. The results are depicted in Figure A.4. The series of experiments consists of 10^3 repetitions for each example, 100 independent replications were used in each repetition. Empirical values of the power $1 - \beta$ are determined by counting rejections during the transient phase for every observation index i separately. The results are depicted in Figure A.4 against $\mathrm{E}[X_\infty] - \mathrm{E}[X_i]$, the difference between the expected value in steady state and the expected value at observation index i. The distance $\mathrm{E}[X_\infty] - \mathrm{E}[X_i]$ is given on the abscissa with increasing values, even though $\mathrm{E}[X_\infty] - \mathrm{E}[X_i]$ is decreasing with growing i in these examples. We can see that the empirical power of $1 - \beta_m$ is tending to 1 for an increasing distance. For small distances it does not tend to 0. This indicates that a Type II error, the error of not rejecting a false H_0, cannot be excluded if the distance gets small. This is the reason why, for example, in the case of a quadratic displacement (see Equation (5.32)) the estimated truncation points tend to be smaller than 100, see Table 5.6 and Figure 5.20, when applying the algorithm of Listing 5.1. This is due to the convergence to the truncation point from below. Better results, deeper in steady state, are obtained by the algorithm of Listing 5.3, where possible candi-



(a) quadratic displacement: Equation (5.32)



(b) ARMA$(5,5)$: Equation (5.33)

Figure A.4: Empirical values of the power $1-\beta$ determined by counting rejections during the transient phase.

dates for truncation points are not only shifted by one observation index but they are growing geometrically with 2^{k-1}.

In this appendix we have assessed the significance level α and the power $1-\beta$ of our implementation of the Anderson-Darling 2-sample test empirically. Results are presented in Table A.2 and Figure A.2. By depicting the percentage of rejections in Figure A.3 for each observation index we get an insight of how a homogeneity test based on multiple comparisons could be implemented. Because the Anderson-Darling 2-sample tests in one step of the algorithm are extremely correlated the conservative approach of demanding that all homogeneity tests have to accept H_0 works well for all our examples, which cover a wide range of possible behaviour of simulation output processes. A recommendation is given how a test can be implemented in the case where the conservative approach is not feasible for the algorithm of Listing 5.1. This problem is very unlikely for the algorithms of Listing 5.2 and Listing 5.3, because here the number of homogeneity tests is constant in every step of the algorithms. Empirical values of the experiment-wide significance level and power are also determined and presented in Figure A.4. These results underline that the conservative approach delivers good results.

A.7 Description of Implemented Software

In the Chapters 4, 5 and 6 simulation experiments were done to demonstrate the capability of the proposed methods and algorithms. For this purpose a software tool was developed which implements the methodology. The programing language C++ is used for implementation to guarantee fast execution time. Object orientation is applied to assure that the source code is easy to understand and maintain. It is straight forward to include extensions. A strict error handling provides the user with detailed information about inadequate use of the software. The standard template library (STL) is used for all kinds of advanced data structures. The software tool is developed for the operating system *Linux*, however, it can

be compiled on an arbitrary operating system if necessary libraries are available. For compilation *Makefile*s are provided. The source code is fully documented by *Doxygen*. Details of the proposed methods are already given in previous chapters. Here, the use of the software and its interfaces are described in brief. Further implementation details can be found in the Technical Report [39-Eic07], which is also available via the Internet pages of the Department of Computer Science & Software Engineering of the University of Canterbury.

A.7.1 Initialisation

The initialisation of the software is done by an external text file, which contains all necessary settings and parameterisation. This will be explained by examples. The initialisation file contains entries in the form of:

```
[MethodID]
Parameter=Value
```

MethodID is an identifier for a method. All following entries are done for this method. If no method is given, e.g. at the beginning of the initialisation file, the entries are regarded to be global. *Parameter* is an identifier which should not contain any special characters. *Value* can be a number or an identifier which is assigned to the connected *Parameter*. Each entry is stated in one line of the initialisation file. All text that follows "//" is regarded as comment and is ignored by the parser. Note, the parser is case sensitive. The initialisation file should contain sensible combination of the following entries:

```
// ***** Global *****
replications=99
resultfile=/home/results.txt
```

These settings are global entries. *Replications* is a necessary parameter which should not be smaller than 3. *Resultfile* is an optional parameter which contains the name of a file. The analyser is using this file to report final results.

```
[deterministic_TPD]
execute=yes // yes, no
cutoff=100 // >0
```

The method *deterministic_TPD* deletes a deterministic number of observation indexes in the beginning of the simulation experiment. The number of deleted observation indexes is given by *cutoff*. The parameter *execute* enables or disables the execution of the method.

```
[sequential_TPD]
execute=yes       // yes, no
ratio=auto        // auto, >0
ratio_min=10      // >0
ratio_max=1000    // >ratio_min
alpha=0.05        // >0 and <1
performance=exact // exact, precise, fast
limit=10000       // >0
```

The method *sequential_TPD* implements the algorithms described in Chapter 5.4. *ratio* can either be a number or *auto*. If *auto* is chosen the parameterisation is done as described in Section 5.5 and *ratio_min* and *ratio_max* have to be defined. *alpha* defines the significance level $1 - \alpha$ of the Anderson-Darling test. *performance* defines which algorithmic approach is used. Choose *exact* for the approach of Listing 5.1, *precise* for the approach of Listing 5.2 or *fast* for the approach of Listing 5.3. *limit* sets the observation index after which the simulation output process is regarded as unstable.

```
[sequential_batching]
execute=yes             // yes, no
independence=vonNeumann // runsUpDown, runsAboveBelow, vonNeumann,
                        // pearsonStrelen, pearsonPermutation
statistic=mean          // mean, spacing
batch_max=100           // >1
sort=yes                // yes, no
alpha=0.05              // >0 and <1
```

The method *sequential_batching* implements non overlapping batching and is used to transform autocorrelated output data into almost independent data. By

independence the test statistic can be defined. Choose *runsUpDown* or *runsAboveBelow* for the run test described in [128-SE43], [123-Sie56], [36-Edi61], [20-Bra68] and [81-Knu98]; choose the setting *vonNeumann* for the test described in [133-vN41] and [52-FY97]; choose *pearsonStrelen* for the test described in Appendix A.1 or choose *pearsonPermutation* for an exact test based on all possible permutations (see e.g. [73-HP36], [107-Pit37], [135-WW43], [136-WW44] and [35-DKS51]). In general, the use of *pearsonPermutation* is not recommendable because of intensive computational effort. *statistic* defines the batch statistic. This can be either *mean* for batch means or *spacing* for using the first observation of each batch. *batch_max* defines the number of used batches. If the parameter *sort* is set to *yes* the data at each observation index is sorted. In this way batching is executed on the order statistics, as needed for the quantile estimator described in Section 6.2. *alpha* defines the significance level $1-\alpha$ of the test for independence.

```
[evolution]
execute=yes // yes, no
start=1     // >0
stop=500    // >start
alpha=0.05  // >0 and <1
```

The method *evolution* implements the method described in Chapter 4. The parameter *start* is an observation index that defines the starting point of the depiction of quantiles and *stop* defines the last depicted observation index. *alpha* sets the confidence level $1-\alpha$ of the confidence interval of the estimated quantiles.

```
[spectral_analysis_QE]
execute=yes // yes, no
batches=128 // >3
alpha=0.05  // >0 and <1
```

The method *spectral_analysis_QE* implements the method of Section 6.2.1. *batches* defines the number of batches and *alpha* defines the confidence level $1-\alpha$ of the confidence interval of the estimated quantiles.

```
[batch_mean_QE]
```

```
execute=yes // yes, no
batches=128 // >3
alpha=0.05  // >0 and <1
```

The method *batch_mean_QE* implements the method described in Section 6.2.2. As in *spectral_analysis_QE*, *batches* defines the number of batches and *alpha* defines the confidence level $1-\alpha$ of the confidence interval of the estimated quantiles.

```
[pooling_QE]
execute=yes       // yes, no
quantiles_min=100 // >0
alpha=0.05        // >0 and <1
```

The method *pooling_QE* implements the method of Section 6.3. *quantiles_min* defines the minimum number of quantiles to be estimated and *alpha* defines the confidence level $1-\alpha$ of the confidence interval of the estimated quantiles. For the methods *spectral_analysis_QE*, *batch_mean_QE* and *pooling_QE* sequential stopping criteria have to be defined. Three different stopping criteria are implemented.

```
[confidenceInterval_SSC_QE]
execute=yes // yes, no
```

confidenceInterval_SSC_QE assures that all estimated quantiles have disjoint confidence intervals.

```
[relativeErrorQuantile_SSC_QE]
execute=yes         // yes, no
critical_value=0.05 // >0 and <1
```

relativeErrorQuantile_SSC_QE assures that the halfwidth of the quantile's confidence interval divided by the quantile estimate is smaller than *critical_value*.

```
[relativeErrorRange_SSC_QE]
execute=yes         // yes, no
critical_value=0.05 // >0 and <1
```

relativeErrorRange_SSC_QE assures that the halfwidth of the quantile's confidence interval divided by the observed range of the distribution is smaller than *critical_value*.

A.7.2 Interface

After compilation the software can be executed by simply giving the name of the initialisation file as the only parameter, e.g.:

```
./analyser_SynchronousMRIP InitialisationFile.init
```

Then, the analyser is started an is waiting for input via the pipe *STDIN_FILENO*. The concept of pipes is part of every modern operating system. One input entity is the observations of all replications at the current observation index as a set of *long double* values in binary form. The number of *long double* values is given by the global parameter *replications*. The following source code is a basic example which shows how to send data to the analyser.

```
std::list<long double> writeMe; // observed data
const size_t LDSize=sizeof(long double);
const int noReplications=writeMe.size();
unsigned char* buffer=0;
long double* ld_ptr=0;

buffer=(unsigned char*)malloc(LDSize*noReplications);
std::list<long double>::const_iterator it=writeMe.begin();
ld_ptr=(long double*)buffer;
for (int i=0;i<noReplications;++i){ld_ptr[i]=(*it);++it;}

int result = write(pipeID, buffer, LDSize*noReplications);
if (result <= 0) return false;
if (result != LDSize*noReplications) throw "error";

free(buffer);buffer = 0;ld_ptr = 0;
```

However, we recommend to use the more advanced routine *send* of the class *interface_multipleRuns*. This routine can be found in the folder *share* in file *interface.cc* and is defined in file *interface.h*.

A.7.3 Simulation Environment

The analyser is the main software module that implements the methods and algorithms which are discussed in previous chapters. It is designed to be part of

a universal simulation controller, like *Akaroa2* ([47-EPM99]), that supports data collection, sequential analysis and stopping simulation when results become satisfactorily accurate. To run the simulation experiments of previous chapters other modules are needed as well. These modules can be regarded as test environment for the analyser and will be discussed in brief.

A simulator is needed to create output data of various simulation processes. Because we operate with multiple replications the pseudo-random number generator has to be coordinated. Therefore, each simulating process receives a number representing which replication it is. On basis of this number the simulation process can pick an exclusive and independent substream of random numbers. The overall seed of the pseudo-random number generator is stored in a file. Once the seed is used for a simulation experiment it is updated to initialise it for the next simulation experiment. The pseudo-random number generator we used is discussed in Section 3.5.

Another module is needed, which coordinates replications, analyser and the flow of data. For the main inter process communication pipes are used. This is a straight forward solution which needs to be extended when distributing replications e.g. in the Internet. To be able to e.g. stop processes in situations where an error occurred *signals* are used. As the concept of *pipes*, the concept of *signals* is part of every modern operating system.

Further modules are implemented for meta analysis (e.g. coverage of interval estimates), to coordinate sequences of simulation experiments or to run just a single simulation run for reasons of comparison with independent replications. These modules are not explained in further detail.

Integration of the quantile analyser into *Akaroa2* is not straight forward, because it follows a different scenario of parallel replications. Replications used by *Akaroa2* run without any synchronisation. They report local estimates at certain checkpoints, which are used to calculate a global estimate. This guarantees

that there is no overhead due to communication between replications. Furthermore, each replication can run at its own speed. This is contrary to the scenario of synchronised replications, which is needed for the methods of this research work. Thus, forcing *Akaroa2* to synchronise its replications will slow them down. However, simulation analysis will be more powerful due to the ability to estimate quantiles.

Bibliography

[1-AS65] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover Publication, Inc., 1965. 119

[2-Ada83] N. R. Adam. Achieving a confidence interval for parameters estimated by simulation. *Management Science*, 29(7):856–866, 1983. 170

[3-AG04] C. Alexopoulos and D. Goldsman. To batch or not to batch. *ACM Transactions on Modeling and Computer Simulation*, 14(1):76–114, January 2004. 87

[4-Amd67] G. Amdahl. Validity of the single processor approach to achieve large scale computing capabilities. *AFIPS Conference Proceedings*, pages 483–485, 1967. 42

[5-AD54] T.W. Anderson and D.A. Darling. A test of goodness of fit. *Journal of the American Statistical Association*, 49:765–769, 1954. 92

[6-AB89] B. C. Arnold and N. Balakrishnan. *Lecture Notes in Statistics: Relations, Bounds and Approximations for Order Statistics*. Springer, 1989. 16

[7-AW95] A. N. Avramidis and J. R. Wilson. Correlation-induction techniques for estimating quantiles in simulation experiments. *Pro-*

ceedings of the 1995 Winter Simulation Conference, pages 268–277, 1995. 30

[8-AW98] A. N. Avramidis and J. R. Wilson. Correlation-induction techniques for estimating quantiles in simulation experiments. *Operations Research*, 46(4):574–591, July-August 1998. 30

[9-AG06] H. P. Awad and P. W. Glynn. On an initial transient deletion rule with rigorous theoretical support. *Proceedings of the 2006 Winter Simulation Conference*, pages 186–191, 2006. 78, 154

[10-Ban98] J. Banks. *Handbook of Simulation*. John Wiley & Sons, Inc., 1998. 11

[11-BCN96] J. Banks, J. S. Carson, and B. L. Nelson. *Discrete-Event System Simulation*. Prentice Hall International, Inc., 1996. 11

[12-BB99] F. Bause and H. Beilner. Intrinsic problems in simulation of logistic networks. *Proceedings of the 11th European Simulation Symposium and Exhibition (ESS99)*, pages 193–198, 1999. 66

[13-BE02] F. Bause and M. Eickhoff. Initial transient period detection using parallel replications. *Proceedings of the 14th European Simulation Symposium*, pages 85–92, 2002. 100

[14-BE03] F. Bause and M. Eickhoff. Truncation point estimation using multiple replications in parallel. *Proceedings of the 2003 Winter Simulation Conference*, pages 414–421, 2003. 78, 94, 100

[15-Ber79] R. Bergmann. Qualitative properties and bounds for the serial covariances of waiting times in single-server queues. *Operations Research*, 27(6):1168–1179, 1979. 60

[16-BA95] G. Berry and P. Armitage. Mid-p confidence intervals: A brief reviem. *The Statistician*, 44(4):417–423, 1995. 23

[17-BEPS07] R. Bolla, M. Eickhoff, K. Pawlikowski, and M. Sciuto. Modeling file popularity in peer-to-peer file sharing systems. *In the Proceedings of the 14th International Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMATA)*, pages 149–155, 2007. 7, 8, 48, 71, 74

[18-BTD06] B. Boltjes, F. Thiele, and I.F. Diaz. Credibility and validation of simulation models for tactical ip networks. *Proceedings of the Military Communications Conference (MILCOM)*, pages 1–10, 2006. 3

[19-BJR94] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis*. Prentice-Hall, Inc., 1994. 236

[20-Bra68] J. V. Bradley. *Distribution-Free Statistical Tests*. Prentice-Hall, Inc., 1968. 262

[21-BFS87] P. Bratley, B. L. Fox, and L. E. Schrage. *A Guide to Simulation*. Springer, 1987. 11

[22-CDL+92] C. R. Cash, D. G. Dipplod, J. M. Long, W. P. Pollard, and B. L. Nelson. Evaluation of tests for initial-condition bias. *Proceedings of the 1992 Winter Simulation Conference*, pages 577–585, 1992. 116, 117, 118

[23-CL99] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999. 11

[24-Che02] E. J. Chen. Two-phase quantile estimation. *Proceedings of the 2002 Winter Simulation Conference*, pages 447–455, 2002. 33, 162

[25-CK99] E. J. Chen and W. D. Kelton. Simulation-based estimation of quantiles. *Proceedings of the 1999 Winter Simulation Conference*, pages 428–434, 1999. 29, 49, 51

[26-CK01] E. J. Chen and W. D. Kelton. Quantile and histogram estimation. *Proceedings of the 2001 Winter Simulation Conference*, pages 451–459, 2001. 33

[27-CK06] E. J. Chen and W. D. Kelton. Quantile and tolerance-interval estimation in simulation. *European Journal of Operations Research*, 168:520–540, 2006. 33

[28-CK06] E. J. Chen and W. D. Kelton. Empirical evaluation of data-based density estimation. *Proceedings of the 2006 Winter Simulation Conference*, pages 333–341, 2006. 33

[29-Con99] W.J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, Inc., New York, 1999. 16, 90

[30-Dan90] W. W. Daniel. *Applied Nonparametric Statistics*. PWS-KENT Publishing Company, 1990. 90

[31-Dar68] D. J. Darley. The serial correlation coefficients of waiting times in a stationary single server queue. *The Journal of the Australian Mathematical Society*, 8(4):683–699, 1968. 112

[32-Dar57] D. A. Darling. The kolmogorov-smirnov, cramer-von mises tests. *The Annals of Mathematical Statistics*, 28(4):823–838, December 1957. 92

[33-DJ54] F. N. David and N. L. Johnson. Statistical treatment of censored data. *Biometrika*, 41(1):228–240, June 1954. 19, 163, 190

[34-Dav70] H. A. David. *Order Statistics*. John Wiley & Sons, Inc., 1970. 16, 20, 21, 162, 163

[35-DKS51] S. T. David, M. G. Kendall, and A. Stuart. Some questions of distribution in the theory of rank correlation. *Biometrika*, 38(1):131–140, June 1951. 262

[36-Edi61] E. S. Edington. Probability table for number of runs of signs of first differences in ordered series. *Journal of the American Statistical Association*, 56:156–159, 1961. 262

[37-Eic02] M. Eickhoff. Statistische Auswertung und Erkennung der stationären Phase in der Simulation zustands-diskreter Systeme. Diploma Thesis, Uni Dortmund, Fachbereich Informatik, 2002. 77

[38-Eic06] M. Eickhoff. Steady state quantile estimation. *Proceedings of the 13th GI/ITG Conference on Measurement, Modeling and Evaluation of Computer and Communication Systems (MMB'06)*, pages 155–171, 2006. 7, 8, 76, 99, 158

[39-Eic07] M. Eickhoff. Sequential quantile estimation. Technical Report TR-COSC 01/07, University of Canterbury, Computer Science & Software Engineering, 2007. 260

[40-EMP05a] M. Eickhoff, D. McNickle, and K. Pawlikowski. Depiction of transient performance measures using quantile estimation. *Proceedings of the 19th European Conference on Modelling and*

Simulation (ECMS'2005), pages 358–363, 2005. 7, 8, 48, 54, 62, 63, 64, 66, 119

[41-EMP05b] M. Eickhoff, D. McNickle, and K. Pawlikowski. Efficient truncation point estimation for arbitrary performance measures. *Proceedings of the 3rd Industrial Simulation Conference (ISC'2005)*, pages 5–12, 2005. 7, 8, 76, 94, 106

[42-EMP06] M. Eickhoff, D. McNickle, and K. Pawlikowski. Analysis of the time evolution of quantiles in simulation. *International Journal of Simulation: Systems, Science & Technology*, 7(6):44–55, 2006. 7, 8, 48, 54, 66, 68, 69, 71

[43-EMP07a] M. Eickhoff, D. McNickle, and K. Pawlikowski. Detecting the duration of initial transient in steady state simulation of arbitrary performance measures. *In the Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools (Valuetools'07)*, 2007. 7, 8, 76

[44-EMP07b] M. Eickhoff, D. McNickle, and K. Pawlikowski. Using parallel replications for sequential estimation of multiple steady state quantiles. *In the Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools (Valuetools'07)*, 2007. 7, 8, 22, 158

[45-EJJ80] R. C. Elandt-Johnson and N. L. Johnson. *Survival Models and Data Analysis*. John Wiley & Sons, Inc., 1980. 90

[46-ES70] J. R. Emshoff and R. L. Sisson. *Design and use of Computer Simulation Models*. The MacMillan Company, 1970. 11

[47-EPM99] G. Ewing, K. Pawlikowski, and D. McNickle. Akaroa-2: Exploiting network computing by distributing stochastic simulation. *Proceedings of the 1999 European Simulation Multiconference*, pages 175–181, 1999. 4, 43, 118, 265

[48-FMG+01] M. J. Fischer, D. M. B. Masi, D. Gross, J. Shortle, and P. H. Brill. Using quantile estimates in simulating internet queues with pareto service times. *Proceedings of the 2001 Winter Simulation Conference*, pages 477–485, 2001. 5, 71

[49-Fis73] G. S. Fishman. *Concepts and Methods in Discrete Event Digital Simulation*. John Wiley & Sons, Inc., 1973. 11, 115

[50-Fis78] G. S. Fishman. Grouping observations in digital simulation. *Management Science*, 24(5):510–521, January 1978. 35, 167, 168

[51-Fis01] G. S. Fishman. *Discrete-Event Simulation*. Springer, 2001. 11

[52-FY97] G. S. Fishman and L. S. Yarberry. An implementation of the batch means method. *INFORMS Journal on Computing*, 9(3):296–310, Summer 1997. 29, 31, 169, 170, 262

[53-Fuj90] R. Fujimoto. Parallel discrete event simulation. *Communications of the ACM*, 33(10):30–53, October 1990. 37

[54-GAM78] A. V. Gafarian, C. J. Ancker, and T. Morisaku. Evaluation of commonly used rules for detecting steady state in computer simulation. *Naval Research Logistics Quarterly*, 25:511–529, 1978. 77, 115

[55-Gho04] B. Ghorbani. The issue of initial transient in sequential steady-state simulation. Master thesis, University of Canterbury, Department of Computer Science, 2004. 145

[56-GC92] J. D. Gibbons and S. Chakraborti. *Nonparametric Statistical Inference*. Marcel Dekker, Inc., 1992. 90

[57-Gly90] P. W. Glynn. Likelihood ration gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1990. 4

[58-GI87] P. W. Glynn and D. L. Iglehart. A new bias deletion rule. *Proceedings of the 1987 Winter Simulation Conference*, pages 318–319, 1987. 78

[59-GH91a] P.W. Glynn and P. Heidelberger. Analysis of initial transient deletion for replicated steady-state simulations. *Operations Research Letters*, 10(8):437–443, 1991. 78

[60-GH91b] P.W. Glynn and P. Heidelberger. Analysis of parallel replicated simulations under a completion time constraint. *ACM Transactions on Modeling and Computer Simulations*, 1(1):3–23, January 1991. 78

[61-GH92a] P.W. Glynn and P. Heidelberger. Analysis of initial transient deletion for parallel steady-state simulations. *Siam J. Scientific. Stat. Computing*, 13(4):904–922, July 1992. 41, 78

[62-GH92b] P.W. Glynn and P. Heidelberger. Experiments with initial transient deletion for parallel, replicated steady-state simulations. *Management Science*, 38(3):400–418, March 1992. 78

[63-GS97] D. Goldsman and B. W. Schmeiser. Computational efficiency of batching methods. *Proceedings of the 1997 Winter Simulation Conference*, pages 202–207, 1997. 28

[64-GSS94] D. Goldsman, L. W. Schruben, and J. J. Swain. Tests for transient means in simulated time series. *Naval Research Logistics*, 41:171–187, 1994. 116, 117, 145

[65-Gor69] G. Gordon. *System Simulation*. Prentice-Hall, 1969. 11

[66-GH98] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory*. John Wiley & Sons, Inc., 1998. 246, 247, 250

[67-Ham94] J. D. Hamilton. *Time Series Analysis*. Princeton University Press, 1994. 61, 235

[68-HS94] S. Hashem and B. W. Schmeiser. Algorithm 727 quantile estimation using overlapping batch statistics. *ACM Transactions on Mathematical Software*, 20(1):100–102, March 1994. 32

[69-Hei86] P. Heidelberger. Statistical analysis of parallel simulations. *Proceedings of the 1986 Winter Simulation Conference*, pages 209–295, 1986. 40

[70-Hei88] P. Heidelberger. Discrete event simulations and parallel processing: Statistical properties. *SIAM Journal of Statistical Computation*, 9(6):1114–1132, 1988. 40, 44

[71-HL84] P. Heidelberger and P.A.W. Lewis. Quantile estimation in dependent sequences. *Operations Research*, 32(1):185–209, February 1984. 29, 30, 31, 32, 164, 231

[72-HW81] P. Heidelberger and P. D. Welch. A spectral method for confidence intervall generation and run length control in simulations. *Communications of the ACM*, 24(4):233–245, April 1981. 29, 31, 35, 117, 164, 166

[73-HP36] H. Hotelling and M. R. Pabst. Rank correlation and tests of significance involving no assumption of normality. *The Annals of Mathematical Statistics*, 7(1):29–43, 1936. 262

[74-Ige76] D. L. Iglehart. Simulating stable stochastic systems, vi: Quantile estimation. *Journal of the Association for Computer Machinery*, 23(2):347–360, April 1976. 5, 29

[75-Jai91] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, Inc., 1991. 11, 81, 88, 200, 243

[76-JC85] R. Jain and I. Chlamtac. The P^2 algorithm for dynamic calculations of quantiles and histograms without storing observations. *Communications of the ACM*, 28(10):1076–1085, October 1985. 5, 28, 29

[77-JFX03] X. Jin, M. C. Fu, and X. Xiong. Probabilistic error bounds for simulation quantile estimators. *Management Science*, 14(2):230–246, February 2003. 5, 30

[78-Kel85] W. D. Kelton. Transient exponential-erlang queues and steady-state simulation. *Communications of the ACM*, 28:741–749, 1985. 248

[79-KL85] W. D. Kelton and A. M. Law. The transient behavior of the m/m/s queue, with implications for steady-state simulation. *Operations Research*, 33:378–396, 1985. 81, 85, 88, 243, 244

[80-Ken40] M. G. Kendall. Note on the distribution of quantiles for large samples. *Supplement to the Journal of the Royal Statistical Sociciety*, 7(1):83–85, 1940. 22

[81-Knu98] D. E. Knuth. *The Art of Computer Programming*. Addison Wesley, 1998. 33, 262

[82-Kol41] A. N. Kolmogorov. Confidence limits for an unknown distribution function. *Annals of Mathematical Statistics*, 12(4):461–463, 1941. 91

[83-KCC05] S. Kurkowski, T. Camp, and M. Colagrosso. Manet simulation studies: The incredibles. *ACM's Mobile Computing and Communications Review*, 9(4):50–61, 2005. 3

[84-LC79] A. M. Law and J. S. Carson. A sequential procedure for determining the length of a steady-state simulation. *Operations Research*, 27(5):1011–1025, September-October 1979. 169, 234

[85-LK00] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, New York, 2000. 3, 11, 13, 39, 160

[86-LS07] P. L'Ecuyer and R. Simard. TestU01: A C library for empirical testing of random number generators. *ACM Transactions on Mathematical Software*, 33(4):Article 22, August 2007. 46

[87-LSCK02] P. L'Ecuyer, R. Simard, E. J. Chen, and W. D. Kelton. An object-oriented random-number package with many long streams and substreams. *Operations Research*, 50(6):1073–1075, 2002. 46, 56, 119

[88-LG89] A. Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addision-Wesley, 1989. 80

[89-Lin95] Y.B. Lin. Parallel independent replicated simulation on a network of workstations. *Simulation*, 64(2):102–110, 1995. 40

[90-LH02] J. R. Linton and C. M. Harmonosky. A comparison of selective initialization bias elimination methods. *Proceedings of the 2002 Winter Simulation Conference*, pages 1951–1957, 2002. 77

[91-MI04] P. S. Mahajan and R. G. Ingalls. Evaluation of methods used to detect warm-up period in steady state simulation. *Proceedings of the 2004 Winter Simulation Conference*, pages 663–671, 2004. 77

[92-McN91] D. McNickle. Estimating the average delay of the first n customers in an m/erlang/1 queue. *Asia-Pacific Journal of Operational Research*, 8:44–54, 1991. 81, 248

[93-MEP04] D. McNickle, G. Ewing, and K. Pawlikowski. Refining spectral analysis for confidence interval estimation in sequential simulation. *Proceedings of the 16th European Simulation Symposium*, 2004. 117, 166

[94-MF00] Z. Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer, 2000. 238

[95-Mih72] G. A. Mihram. *Simulation: Statistical Foundations and Methodology*. Academic Press, 1972. 11

[96-NW88] H. R. Neave and P. L. Worthington. *Distribution Free Tests*. Unwin Hyman Ltd, 1988. 90

[97-New98] R. G. Newcombe. Two-sided confidence intervals for the single proportian: Comparison of seven methods. *Statistics in Medicine*, 17:857–872, 1998. 51

[98-Pap84] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1984. 80

[99-Paw90] K. Pawlikowski. Steady-state simulation of queueing processes: a survey of problems and solutions. *ACM Computing Surveys*, 22:123–170, June 1990. 3, 4, 77, 115, 167, 169, 170

[100-Paw00] K. Pawlikowski. Distributed stochastic discrete-event simulation. *Simulation in Research and Development*, pages 8–17, 2000. 41

[101-PJL02] K. Pawlikowski, H-D. J. Jeong, and J-S. R. Lee. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine*, 40(1):132–139, 2002. 3

[102-PM01] K. Pawlikowski and D. McNickle. Speeding up stochastic discrete-event simulation. *Proceedings of the 13th European Simulation Symposium (ESS01)*, October 2001. 42

[103-PME98] K. Pawlikowski, D. McNickle, and G. Ewing. Coverage of confidence intervals in sequential steady-state simulation. *Journal of Simulation Practise and Theory*, 6(3):255–267, 1998. 186

[104-PSM06] K. Pawlikowski, M. Schoo, and D. McNickle. Modern generators of multiple streams of pseudo-random numbers. *Proceedings of the International Mediterranean Modelling Multiconference (ESM06)*, pages 553–559, 2006. 46

[105-PYM94] K. Pawlikowski, V.W. Yau, and D. McNickle. Distributed stochastic discrete-event simulation in parallel time streams. *Proceedings of the 1994 Winter Simulation Conference*, pages 723–730, 1994. 41

[106-Pet76] A. N. Pettitt. A two-sample anderson-darling rank statistik. *Biometrika*, 63(1):161–168, 1976. 92

[107-Pit37] E. J. G. Pitman. Significance tests which may be applied to samples from any populations. ii. the correlation coefficient test. *Supplement to the Journal of the Royal Statistical Society*, 4(2):225–232, 1937. 234, 262

[108-Raa87] K. E. E. Raatikainen. Simultaneous estimation of several percentiles. *SIMULATION*, 49(4):159–164, October 1987. 32

[109-Raa90] K. E. E. Raatikainen. Sequential procedure for simultaneous estimation of several percentiles. *Transactions of the Society for Computer Simulation*, 7(1):21–44, 1990. 32

[110-Raa95] K. E. E. Raatikainen. Simulation-based estimation of proportions. *Management Science*, 41(7):1202–1223, July 1995. 32, 34, 35, 159, 160

[111-RW89] R. Righter and J. C. Walrand. Distributed simulation of discrete event systems. *Proceedings of the IEEE*, 77(1):99–113, January 1989. 37

[112-Rob02] S. Robinson. A statistical process control approach for estimating the warm-up period. *Proceedings of the 2002 Winter Simulation Conference*, pages 439–446, 2002. 77

[113-Rob07] S. Robinson. A statistical process control approach to selecting a warm-up period for a discrete-event simulation. *European Journal of Operational Research*, 176(1):332–346, 2007. 77

[114-RLQ05] M. D. Rossetti, Z. Li, and P. Qu. Exploring exponentially weighted moving average control charts to determine the warm-up period. *Proceedings of the 2005 Winter Simulation Conference*, pages 771–780, 2005. 77

[115-SS86] F. W. Scholz and M. A. Stephens. K-sample anderson-darling tests of fit, for continuous and discrete cases. Technical Report 81, Department of Statistics, University of Washington, May 1986. 92

[116-SS87] F. W. Scholz and M. A. Stephens. K-sample anderson-darling tests. *Journal of the American Statistical Association*, 82(399):918–924, September 1987. 92, 93, 98

[117-Sch82] L. W. Schruben. Detecting initialization bias in simulation output. *Operations Research*, 30(3):569–590, May-June 1982. 87, 116

[118-SST83] L. W. Schruben, H. Singh, and L. Tierney. Optimal tests for initialization bias in simulation output. *Operations Research*, 31(6):1167–1178, November-December 1983. 87, 116, 118, 145

[119-Sei82] A. F. Seila. A batching approach to quantile estimation in regenerative simulations. *Management Science*, 28(5):573–581, May 1982. 29

[120-Sha95] P. Shahabuddin. Rare event simulation in stochastic models. In *Proceedings of the 1995 Winter Simulation Conference*, pages 178–185, Arlington, Virginia, USA, 1995. 4

[121-Sha75] R. E. Shannon. *Systems simulation: the art and science*. Prentice-Hall, Inc., 1975. 11

[122-She97] D. J. Sheskin. *Parametric and Nonparametric Statistical Procedures*. CRC Press LLC, 1997. 90

[123-Sie56] S. Siegel. *Nonparametric Statistics*. McGRAW-HILL BOOK COMPANY, INC., 1956. 169, 262

[124-Smi48] N. V. Smirnov. Table for estimating the goodness of fit of empirical distributions. *Annals of Mathematical Statistics*, 19(2):279–281, June 1948. 91

[125-Spr03] J. C. Sprott. *Chaos and Time-Series Analysis*. Oxford University Press, 2003. 69

[126-Ste74] M. A. Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737, September 1974. 90

[127-Str04] J. C. Strelen. The accuracy of a new confidence interval method. *Proceedings of the 2004 Winter Simulation Conference*, pages 654–662, 2004. 17, 233

[128-SE43] F. S. Swed and C. Eisenhart. Tables for testing randomness of grouping in a sequence of alternatives. *The Annals of Mathematical Statistics*, 14(1):66–87, March 1943. 262

[129-Tho36] W. R. Thompson. On confidence ranges for the median and other expectation distributions for populations of unknown distribution form. *The Annals of Mathematical Statistics*, 7(3):122–128, September 1936. 17

[130-Toc63] K. D. Tocher. *The Art of Simulation*. D. Van Nostrand Co., Inc., 1963. 11

[131-Tri02] K. S. Trivedi. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. John Wiley & Sons, Inc., 2002. 14, 80

[132-VAVA94] M. Villén-Altamirano and J. Villén-Altamirano. RESTART: A straightforward method for fast simulation of rare events. In *Proceedings of the 1994 Winter Simulation Conference*, pages 282–289, 1994. 5

[133-vN41] J. von Neumann. Distribution of the ratio of the mean square successive difference to the variance. *The Annals of Mathematical Statistics*, 12(4):367–395, December 1941. 262

[134-Vor02] N. N. Vorobiev. *Fibonacci Numbers*. Birkhäuser Verlag, 2002. 241

[135-WW43] A. Wald and J. Wolfowitz. An exact test for randomness in the non-parametric case based on serial correlation. *The Annals of Mathematical Statistics*, 14(4):378–388, December 1943. 234, 262

[136-WW44] A. Wald and J. Wolfowitz. Statistical tests based on permutations of the observations. *The Annals of Mathematical Statistics*, 15(4):358–372, December 1944. 262

[137-Wel83] P. D. Welch. The statistical analysis of simulation results. *In The Computer Performance Modeling Handbook, ed. S. Lavenberg, Academic Press*, pages 268–328, 1983. 78

[138-Whi91] W. Whitt. The efficiency of one long run versus independent replications in steady-state simulation. *Management Science*, 77(6):645–666, June 1991. 40, 44

[139-WS94] D. C. Wood and B. Schmeiser. Consistency of overlapping batch variances. *Proceedings of the 1994 Winter Simulation Conference*, pages 316–319, 1994. 32

[140-WS95] D. C. Wood and B. Schmeiser. Overlapping batch quantiles. *Proceedings of the 1995 Winter Simulation Conference*, pages 303–307, 1995. 19

[141-ZW07] J. Zhang and Y. Wu. k-sample tests based on the likelihood rati. *Computational Statistics & Data Analysis*, 51:4682–4691, 2007. 96

Note that the numbers, which are stated behind every item of the bibliography, refer to the pages where they are used.