Statistical Analysis of Software Reliability Models

Statistical Analysis of Software Reliability Models

Statistische analyse van software reliability modellen
(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor
aan de Rijksuniversiteit te Utrecht
op gezag van de Rector Magnificus, Prof. Dr. J. A. van Ginkel,
ingevolge het besluit van het College van Dekanen
in het openbaar te verdedigen
op woensdag 24 februari 1993 des namiddags te 2.30 uur

door

Marcus Cornelus Jacobus van Pul

geboren op 30 april 1966
te Oss

Promotor: Prof. Dr. R.D. Gill *(Faculteit der Wiskunde en Informatica)*
Co-promotor: Dr. K. Dzhaparidze *(Centrum voor Wiskunde en Informatica, Amsterdam)*

Het onderzoek voor dit proefschrift werd gefinancierd door de Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) via de Stichting voor de Technische Wetenschappen (STW) in het kader van project CWI 88.1481 "Statistical Analysis of Error-counting and Debugging Models in Software Reliability".

ISBN 90-393-0265-0

Preface

The past two decades have witnessed an enormous growth of literature on software reliability theory. The preoccupation with model-building has resulted in a large number of theoretical models which seem to lack immediate appeal from a statistical point of view. The aim of this thesis is to attempt to reduce the gap between theory and applications in the area of software reliability.

This thesis has been divided into three parts: introduction, theory and applications. The introductory Part I consists of two chapters. In Chapter 1 a general introduction to software reliability is given. We discuss some basic concepts, models and assumptions and describe some different approaches to the subject. Chapter 2 provides most of the mathematical framework. The relevant notions of reliability theory and counting process theory are presented. In the last section of this chapter we give an overview of the theoretical results of this thesis. Part II consists of five technical papers which recently appeared in scientific journals or will appear fairly soon. As we have chosen to present the papers in the form they are published (or will be published) there will be some duplication in the introductory and framework sections. When we found it appropriate to add a special note, not present in the original publication, we did this between square brackets and in a slightly smaller type font. [This will look like this.] Chapter 3 studies the asymptotic properties of maximum likelihood estimators for a broad class of models. In Chapter 4 the validity of the parametric bootstrap is derived in our situation and simulation results are investigated for a particular software reliability model. In Chapter 5, which consists of joint work with Leo G. Barendregt, for one of the more popular software reliability models, solutions are provided to the classical statistical problem of multiple solutions to the system of likelihood equations. Chapter 6 is devoted to the derivation of the limit distribution of some goodness of fit test statistics, while in Chapter 7 a new, more sophisticated software reliability model is introduced and analysed. In Part III of this thesis we consider applications. In Chapter 8 the results of two software reliability case-studies at Philips Medical Systems (PMS) and Ericsson Telecommunications (ETM) are presented. I would like to thank Tom Hoogenboom, Johan van Beers and Jelle Rieske (all of PMS) and Peter Westeneng and Han van Beek (both of ETM) for their co-operation, patience and careful reading. Finally, in an appendix a software quality tool, implemented in S-PLUS, is demonstrated.

The research work that is presented in this thesis, is the result of the project *Statistical Analysis of Error-Counting and Debugging Models in Software Reliability* that was supported by the Dutch Technology Foundation (STW) and was carried out at the Centre for Mathematics and Computer Science (CWI) in Amsterdam. I would like to thank both organisations for the generous support that I have received. A large number of people have contributed in one way or an other to the successful completion of this thesis. Although I cannot mention them all personally, they should know I am very indebted for their support, comments or stimulating sympathy. Finally, it has been a great privilege for me to have Richard D. Gill as supervisor. His fabulous mathematical expertise and warm encouragement kept me, whenever overwhelmed by darkness, on the right track.

Contents

PART I
INTRODUCTION

PART II
THEORY

PART III
PRACTICAL APPLICATION

Part I

Introduction

Chapter 1

General Introduction to Software Reliability

1.1 Introduction

To-day it is hard to think of any area in modern society in which computer systems do not play a dominant role. In space- and air-navigation, defence, telecommunication and health-care, to name a few, computers have taken over the most life-critical tasks. Unlike most human beings, computers seem to do their job perfectly, at all times and under all conditions. But do they really? Well, most of the time, they do. Sometimes, however, for some reason a zillion dollar satellite goes off course, a patriot rocket misses its target or a large telephone exchange gives up. Possible sources for such dissatisfactory behaviour are physical deterioration or design faults in hardware components. In the fifties and sixties a general reliability theory was built for hardware. Another source for malfunctioning of computer systems is the presence of bugs in the software that controls the system. A beginning with the modelling of the reliability of software was only made in the early seventies.

Obviously, also in the case of less delicate computer applications, all customers want a high degree of reliability to be guaranteed. Of course, every software-house claims to design and produce software in such a structured and sophisticated way that the result is a perfect computer program. As in general the logical complexity of software is much larger than that of hardware, proving the correctness of a piece of software is in most cases an impossible task. Software developers have to admit that in practice a completed program is never perfect, but, more likely, still full of bugs. Therefore, the software is tested intensively for quite a span of time before it is finally released. Here a difficult trade-off occurs between costs and schedule on the one side and quality on the other. The test time, which can mount up to more than a third of the total development time, seems not productive and therefore extremely expensive. Besides, there exists the risk, that a competitor will release the same product a bit earlier. On the other hand, the sales of an unreliable product will be disappointing and can do more bad than good to the image of the software-house. It seems to make sense to study the evolution of the reliability of computer software during the test and development phase: does the software already satisfy certain criteria or how long should testing be continued?

In this thesis mathematical and statistical aspects of software reliability theory are presented. This first chapter provides a general introduction to software reliability theory. Two excellent handbooks on this topic are Musa *et al.* (1987), and Rook (1990). In the next section we discuss how software reliability relates to hardware reliability. In Section 1.3 some important features related to software reliability are defined. In Section 1.4 we describe several approaches - both static and dynamic - to estimation or prediction of the reliability of a piece of software. The dynamic approach of counting process models will be used throughout this thesis. Classical assumptions and models in this approach are discussed in Section 1.5. Finally, in Section 1.6 we briefly summarise the mathematical and statistical aspects involved. More backgrounds on statistical concepts and techniques can be found in Chapter 2, where we also give an overview of the most important results and conclusions of the main part (Part II) of this thesis.

1.2 Software versus hardware

The field of hardware reliability has been established for some time. Some useful references are: Shooman (1968), Mann *et al.* (1974), Barlow & Proschan (1975), Lawless (1982), and Ascher & Feingold (1984). One might ask how software reliability relates to it. In reality, the division between hardware and software reality is somewhat artificial. Both may be defined in the same way. Therefore, you may combine hardware and software reliability to get system reliability. Both depend on the environment.

The source of failures in software is design faults, while the main source in hardware has generally been physical deterioration. However, the models and methods developed for software reliability could really be applied to any design activity, including hardware design. Once a software defect is properly fixed, it is in general fixed for all time. Failures usually occur only when a program is exposed to an environment that it was not developed or tested for. Software reliability tends to change continuously during test periods, due to the addition of problems in new code or due to the removal of problems by repair actions. Hardware reliability has, apart perhaps from an initial burn-in or end of useful life period, a much greater tendency towards a constant value.

Also in hardware the presence of design faults is possible, but the design reliability concept has not been applied to hardware to any extent. The probability of failure due to wear and other physical causes has usually been much greater than that due to an unrecognised design problem. It was possible to keep hardware design failures low because hardware was generally less complex logically than software. Hardware design failures had to be kept low because retrofitting of manufactured items in the field was very expensive. The emphasis in hardware reliability is starting to change now, however. Awareness of the work that is going on in software reliability, plus a growing realisation of the importance of design faults may be having an effect.

Despite the forgoing differences, we can develop software reliability theory in a way that is compatible with hardware reliability theory. Thus, system reliability figures may be computed using standard hardware combinatorial techniques (Shooman (1968), Lloyd &

Lipow (1977)). In this thesis attention is restricted to problems related to the modelling of the reliability of software only.

1.3 Basic concepts

First of all, we have to make clear what we mean, when using vague terms like software faults and software reliability. In this Section we give some intuitive descriptions; more formal definitions and mathematical expressions are given in Section 2.2 and 2.3. We speak of a *software failure,* if the program-output deviates from what it should be according to the customer. This means that also errors in the specification can lead to software failures. A failure is a dynamic thing; the program has to be executed to detect software failures. A *software fault* (or *bug*) is an error in the program source-text, which when the program is executed under certain conditions can cause a software failure. A software fault is hence generated at the moment a programmer or system analyst makes a mistake. Often one defines the *reliability* of a piece of software as the probability of failure-free execution of the software for a specified time in a specified environment. An operating system, for instance, with a reliability of 95% for 8 hours for an average user, should work 95 out of 100 periods of 8 hours without problems. A characteristic that is strongly correlated with the reliability is the *expected failure intensity,* sometimes called the *rate of occurrence of failures (ROCOF),* which is defined as the expected number of occurring failures per unit of time. Strictly speaking, the expected failure intensity is the first derivative of the *mean value function* which represents the expected cumulative number of failures detected up to each point in time. Between the terms software fault and software failure there exists a cause and effect relation. The terms software reliability and failure intensity both give a measure for the quality of the software. Other interesting measures, which are directly related to these two, are the *mean time between failures (MTBF)* and the *time to release (TTR).*

1.4 Different approaches

Since the early seventies, many researchers have paid attention to the problem of estimation and prediction of software reliability. They used various starting-points, assumptions, and techniques; all aiming at the same goal. In this section we briefly discuss four different approaches to software reliability.

(i) Fault seeding. One can estimate the number of inherent faults in software programs by an empirical method, variously called fault seeding, error seeding or "bebugging" (Mills (1972), Basin (1973), Gilb (1977) and Rudner (1977)). The test-leader introduces a certain number of artificial faults into the program in some suitable random fashion, unknown to the people who will test the software. It is assumed that these seeded faults are equivalent to the inherent faults in terms of difficulty of detection. Inherent and seeded faults discovered are counted separately. The number of inherent faults can be predicted by using the observed proportion of seeded faults found to total seeded faults. The reasoning is based on the concept that with equal difficulty of discovery, the same proportions of both types of faults will have been discovered at any point in time. Unfortunately, it has proved difficult to implement seeding in practice. It is not easy to introduce artificial faults that are equivalent to inherent faults in difficulty of discovery. In general, it is much easier to find

the seeded faults. Consequently, the number of inherent faults is usually underestimated with this technique.

(ii) Iterated testing. An other approach is suggested in Nagel & Scrivan (1982) and Nagel *et al.* (1984). They investigate software reliability by what they call replicated testing or repetitive run experimentation. The idea is to test a large number of identical copies of a software program simultaneously and independently of each other. In this way one can calculate the average number of bugs found as a function of time very precisely. This function can be used for the prediction of the number of bugs that will be found before a certain point in time in the future. There exists, however, a stronger motivation for this approach. Obviously, there will be larger and smaller faults in the software program. Larger faults tend to cause software failures earlier than the smaller ones. A consequence of this is that the larger faults tend to be present in most of the test runs, while smaller faults occur only in a few of them. The key issue of this method is that one can obtain a good empirical estimation of the distribution of the occurrence rates of the different bugs. This information enables one to model the growth of reliability in an appropriate way. A large disadvantage of this approach, however, is that to be of any use iterated testing is enormously time consuming and therefore rarely used in practical situations.

(iii) Static complexity analysis. Following a completely other direction, one can estimate the number of faults from program characteristics only. It has long be assumed that the size of a program has the most effect on the number of inherent faults it contains. Akiyama (1971), Thayer (1976), Motley & Brooks (1977) and Feuer & Fowlkes (1979) have verified this hypothesis. They all more or less consider models in which the number of inherent faults is proportional to some power of the program length. Possibly, other measures of program complexity than just program length can improve the prediction of number of inherent faults. Complexity measures form an active current research area. Very popular are *McCabe's cyclomatic number* (McCabe (1976)) and *Halstead's effort* (Halstead (1977)). Other metrics can be found in Grady & Caswell (1987). However, most of the complexity metrics developed to date show a high correlation with program size. They provide little improvement over just program size alone in predicting inherent faults remaining at the start of system test (Sunohara *et al.* (1981)).

(iv) Error-counting and debugging models. We consider the following experiment. A computer program is tested for a specified length of time. Inputs are selected randomly from the input-space, that is, in a way that is representative for the operational profile. Either the program produces the correct output, or a software failure occurs. That is, the software produces the wrong answer or no answer at all. After the detection of a failure, the CPU-clock is stopped and the program is sent to a team of debuggers. When the fault is found and fixed, available data concerning fault and failure are gathered in a database. After this, the CPU-clock is started again and testing continues with a new input until the end of the test period is reached. See Figure 1.1. Among others, the following data-items are of interest:

- The *failure times,* times at which the failures occur. This could be measured in seconds CPU-time, days real-time or even by the sequence number of the test input.

- A description of the failure (or of the *priority* of its effects), so that a classification of the effects of errors is possible. It can be of use to distinguish failure intensities of different types of failures.

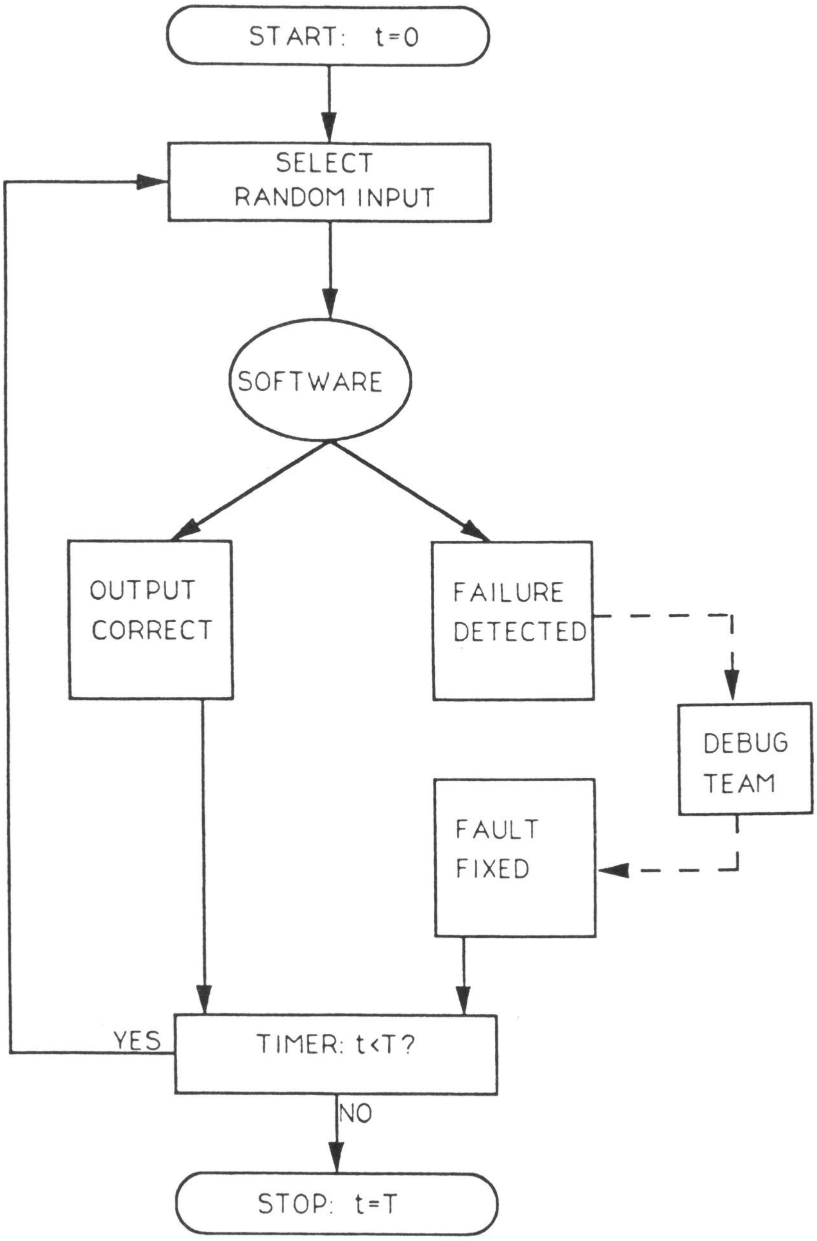


Figure 1.1
The test experiment for error-counting and debugging models.

- A description of the fault (the *cause,* like errors in specification, design or code). Also here a classification can make sense. Certain kinds of faults can indicate for instance a lack of accuracy or knowledge of the programmer or the misinterpretation of possibly ill-posed specifications. One should try to prevent such systematic errors in future.

- The *location* of the fault in the source text. When the program consists of a number of modules, it is possible to find local differences in the failure intensity and hence to concentrate the testing effort in the most critical regions.

- A measure of the *size of the correction* of a fault (for example in bytes or new lines added or changed code or in man-hours). When modelling imperfect repair (that is, there exists a positive chance of introducing new faults when repairing an old one) such information would be very useful.

Only the failure times are essential to be able to conclude something about the evolution of the reliability during the testing process. The class of *Error-Counting and Debugging Models* consists of relatively simple models, considering the test experiment as described above, characterised by the fact that they are only based on certain test data, such as the occurrence times of failures. These error-counting and debugging models do not explicitly depend on factors like the length and the structure of the program, the language in which it is written, the skill of the programmer, etcetera. By using the information obtained from the experiment one can estimate the parameters of the underlying model, in particular the total number of faults initially present in the software. Certain functions of these model-parameters will yield estimates of other interesting quantities (such as the failure intensity, the reliability, the mean time between failures and the release time). In practice, however, decisions about when to stop testing are rarely based solely on critical values for such quantities. More often to find an optimal stopping time, the reliability model is extended by associating cost functions, modelling the cost of testing versus the costs of faults in the field. An optimal stopping rule will tell to stop testing as soon as the cost of discovering and fixing the remaining faults is greater than the cost of repairs in the field.

As stated earlier, approaches (i) and (ii) are nice theoretical concepts, but they have considerable disadvantages that make them less practical. The static approach (iii) and the dynamic approach (iv) are both used in practice. In this thesis we restrict ourselves mainly to the mathematical and statistical aspects of approach (iv), that is of error-counting and debugging models.

1.5 Assumptions and models

Efforts in describing the evolution of the reliability of computer software during testing resulted in the proposal of dozens of error-counting and debugging models over the past twenty years. Each individual model is completely characterised by a certain set of assumptions. Sometimes, we assume that failures in the software will occur independently and that when a failure is detected, the fault is fixed immediately with no new faults introduced. This is the case for some very well-known models: the Jelinski-Moranda model (Jelinski & Moranda (1972)), the Goel-Okumoto model (Goel & Okumoto (1979)) and the Littlewood model (Littlewood (1980)).

The *Jelinski-Moranda model* is the oldest and one of the most elementary software reliability models introduced so far. In this model the failure rate of the program is at any time proportional to the number of remaining faults and each fault makes the same contribution to the failure rate. In the *Goel-Okumoto model* the failures occur according to a non-homogeneous Poisson process. The failure rate does not depend on the debugging process; it is a simple deterministic function which decreases exponentially in time. Both the Jelinski-Moranda model and the Goel-Okumoto model are in some sense special cases of a more general model, the *Littlewood model.* The main difference with respect to the two previous models is the fact that Littlewood does not assume that each fault makes the same contribution to the failure intensity. He allows each fault to have its own occurrence probability. Littlewood's argument for this is that larger faults will produce failures earlier than small ones. The way, however, in which Littlewood assigns occurrence rates to the faults, is rather *ad hoc.* In practice, it turns out that for many data-sets estimates based on Littlewood's model are not better than those based on the Jelinski-Moranda model.

Assumptions like independence of the occurring faults, negligible repair time and perfect repair are of course, not very realistic. It is unknown how large the influence on the results is of such an assumption as the independence. Without this assumption, however, the mathematical problem becomes a lot more complicated. With respect to the assumption of negligible repair time (that is: stop CPU-clock when failure detected) one can add that there are ways of transforming execution-time models into real-time models (Musa (1975)). Moreover, immediate repair is not essential if we take care to count failures due to the same software fault only once. A new and interesting idea seems to be the modelling of imperfect repair and software growth simultaneously. With software growth we mean the phenomenon that a piece of software is not a static object, but on the contrary changes in time. Not only does each repair cause a change in the software, but also in practice at certain moments in the testing phase we will add new modules to the software as well. In the *Poisson Growth & Imperfect Repair-model,* (Van Pul (1991b)), it is assumed that the expected number of new faults introduced at a certain point in time, is proportional to the size of the change in the software at that moment. This assumption makes it possible to model imperfect repair and software growth simultaneously. Besides, the model will account for dependencies between faults.

A more formal treatment of these software reliability models will be given in Section 2.6 after we have introduced the necessary mathematical notations and concepts. Finally, for a complete chronological catalogue of the most popular software reliability models introduced since 1972 we refer to Musa *et al.* (1987).

1.6 Mathematical and statistical aspects

An important statistical problem is the comparison of different models (see Chapter 6 and 7 of this thesis). This is usually done by goodness of fit testing. The test statistics involved are in our situation rather complex, and the derivation of their distributions can cause considerable difficulties. Not only does the choice of the best model confront us with many questions, but also the estimation of the parameters in the chosen model is a difficult problem. We usually use the maximum likelihood estimation procedure for this purpose. We derive the chance (likelihood) to get the data under the parameters and maximise this likelihood as a function of the parameters. The derivation of the likelihood function is not always possible analytically and the numerical computation of its maxima can be unstable.

In practice we will use iterative techniques like the *Newton-Raphson procedure* (Carnahan & Wilkes (1973)) or the *down-hill simplex method* (Nelder & Mead (1965)) to approximate the roots of a system of equations. Chapters 3, 4 and 5 consider the problem of parameter estimation. In the next chapter, Chapter 2, we give some further relevant mathematical background and discuss the above mentioned statistical concepts in more detail.

Chapter 2

Mathematical Framework

2.1 Introduction

The aim of this chapter is to provide some mathematical background to topics that will be important later on. In some cases we confined ourselves to giving references to relevant literature. In the next section we briefly summarise the main concepts in reliability theory. In Section 2.3 we present some important counting process theory results. We shed some light on the way we treat asymptotics and limit theory in Section 2.4. In Section 2.5 we discuss some of the main statistical problems in parameter estimation. We illustrate the theoretical concepts with some examples and place the software reliability models, mentioned in Chapter 1, in a mathematical context in Section 2.6. Finally, in Section 2.7 we give an overview of the theoretical results that are presented in Part II of this thesis.

2.2 General reliability concepts

A reliability study is concerned with random occurrences of undesirable events, or failures, during the life of a physical system. As a failure phenomenon is stochastic by nature, reliability theory heavily depends on probability concepts. To illustrate this, the *reliability* of a physical system will be defined (later on in this section) as the probability that the system performs its task adequately for a specified time in a specified environment.

Let T be a continuously distributed random variable representing the *failure time* (or lifetime) of a physical system. The probability that the system will have failed by time t is

$$
\begin{aligned}
F(t) &:= \mathrm{I\!P}\,(\,T \le t\,) \\
&= \int_0^t f(u)\,\mathrm{d}u,
\end{aligned}
$$

where $F(t)$ and $f(t)$ denote the cumulative distribution function and probability density function (or the failure density function), respectively. The probability of the system surviving until time t is

$$
\begin{aligned}
S(t) &:= \mathrm{I\!P}\,(\,T > t\,) \\
&= 1 - F(t) \\
&= \int_t^\infty f(u)\,\mathrm{d}u,
\end{aligned}
$$

where $S(t)$ is the *survival function.* The *expected life*, or the expected time during which the system will function properly, is defined as

$$
\begin{aligned}
\mathrm{I\!E}\,(\,T\,) &:= \int_0^\infty u\, f(u)\,\mathrm{d}u \\
&= \int_0^\infty S(u)\,\mathrm{d}u.
\end{aligned}
$$

When the system being under consideration is returned to an as good as new state after each repair, $\mathrm{I\!E}\,(\,T\,)$ is also known as the *mean time to failure* (MTTF). Thus the time intervals between successive failure are independent and identically distributed random variables. The renewal process of an electric light bulb is a good example of this. The *failure rate* or *hazard rate* is the rate at which failures occur at time t given that the system survives up to t. Or more formally:

$$
\begin{aligned}
z(t) &:= \lim_{\Delta t \to 0} \frac{F(t+\Delta t) - F(t)}{\Delta t\, S(t)} \\
&= \frac{f(t)}{S(t)} \\
&= \frac{f(t)}{1 - F(t)}.
\end{aligned}
$$

The functions $f(t)$, $S(t)$ and $z(t)$ give mathematically equivalent specifications of the distribution of T. Expressions for $S(t)$ and $f(t)$ are given in term of $z(t)$ by

$$S(t) = \exp\left[-\int_0^t z(u)\,\mathrm{d}u\right]$$

and

$$f(t) = z(t)\exp\left[-\int_0^t z(u)\,\mathrm{d}u\right],$$

respectively. Finally, defining the *reliability* of the system at time t as the probability that it survives another period τ after t, given the system survives up to time t, we have the following mathematical expression:

$$\begin{aligned} R(t,\tau) &:= \mathrm{I\!P}(T > t+\tau \mid T > t) \\ &= \frac{S(t+\tau)}{S(t)} \\ &= \exp\left[-\int_t^{t+\tau} z(u)\,\mathrm{d}u\right]. \end{aligned}$$

The *mean residual life* is given by

$$\mu(t) := \int_0^\infty \frac{S(t+u)}{S(t)}\,\mathrm{d}u.$$

The hazard rate will change over the lifetime of a physical system, due to infant mortality (low ages) and wear-out (high ages). The hazard rate function often has the shape of a so-called bathtub curve, see Figure 2.1. For the cumulative distribution function $F(t)$, distributions such as the gamma and Weibull distributions are often used to represent infant mortality (region I) and wear out (region III), because of their great adaptability. The exponential distribution, which is widely used in reliability because of its great simplicity and applicability, is valid only for region II (that is, during the useful life period or normal operating phase). Note that there exist no direct analogous concepts to infant mortality and wear-out in software reliability theory.

We did not intend to provide a comprehensive treatment of reliability theory. For more in-depth material on this subject we refer to: Shooman (1968), Mann *et al.* (1974), Barlow & Proschan (1975), Lawless (1982) and Ascher & Feingold (1984).

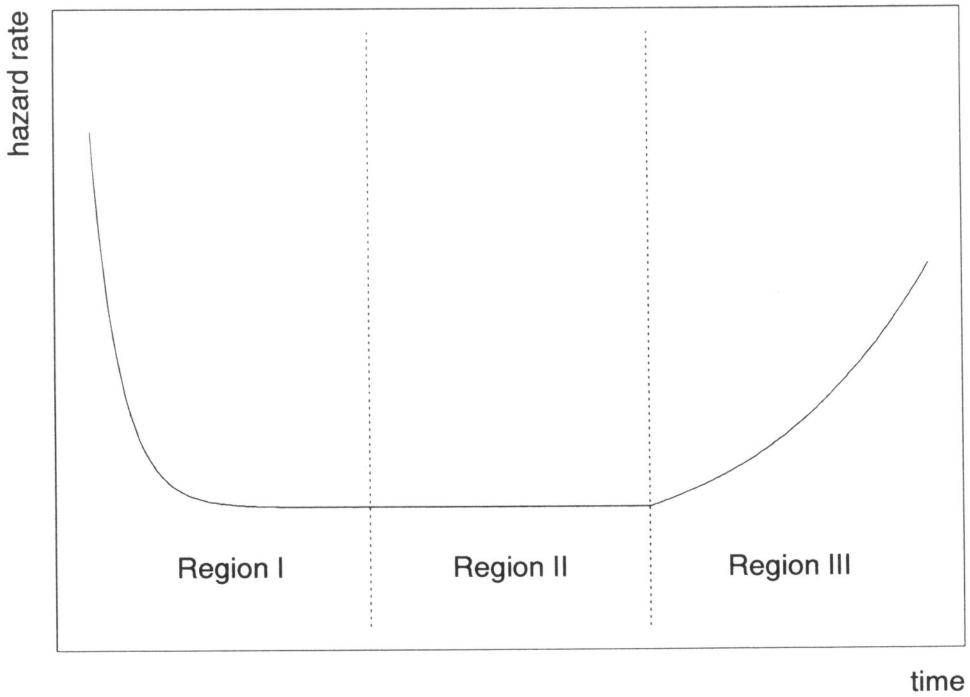


Figure 2.1
Typical hazard rate of an electronic component as a function of age.

2.3 Counting process theory

We are going to model the occurrence of discrete, random events in continuous time. We fix $\mathcal{T} = [0,\tau]$ for a given finite terminal time τ, $0<\tau<\infty$. Recalling approach (iv) of Section 1.4, note that we are observing a non-deterministic process through the fixed time window $\mathcal{T}$. The fact that the number of faults detected in $\mathcal{T}$ will be stochastic is the reason why we cannot use classical maximum likelihood theory for i.i.d. observations in deriving asymptotic results. Therefore we introduce a powerful mathematical instrument which we will use to solve these problems: the theory of counting processes and martingales. For a complete summary we refer to Andersen & Borgan (1985), Jacod & Shiryaev (1987) or Andersen *et al.* (1993). Before we are able to introduce the important notions of martingales, counting processes and their intensities, we have to give some other definitions first.

Let $(\Omega,\mathcal{F},\mathbb{P})$ be a probability space. A *filtration* or *history* $(\mathcal{F}_t : t\in\mathcal{T})$ is an increasing, right-continuous family of sub σ-algebras of $\mathcal{F}$. The σ-algebra $\mathcal{F}_t$ is interpreted as follows: it contains all events whose occurrence or not is fixed by time t. We write correspondingly $\mathcal{F}_{t-}$ for the available data just before time t. A *stochastic process* X is just a time-indexed collection of random variables $(X(t) : t\in\mathcal{T})$. The process X is called *adapted* to the filtration if $X(t)$ is $\mathcal{F}_t$-measurable for each t and *cadlag* if its sample paths $(X(t,\omega) : t\in\mathcal{T})$ for almost all ω are right continuous with left-hand limits. The set of all cadlag functions on $\mathcal{T}$ is often denoted by $D(\mathcal{T})$, the Skohorod space of weak convergence theory. See Billingsley (1968). The *self-exciting* filtration $\mathcal{F}_t$ of a stochastic process X is the σ-algebra generated by $X(s), s \leq t$. Finally, a stochastic process X is called *integrable* if for all $t \in \mathcal{T}$

$$\mathrm{I\!E}\left[\,|X(t)|\,\right] < \infty$$

and *predictable* if as a function of $(t,\omega) \in \mathcal{T} \times \Omega$, it is measurable with respect to the σ-algebra on $\mathcal{T} \times \Omega$ generated by the left continuous adapted processes.

So suppose a filtration $(\mathcal{F}_t : t \in \mathcal{T})$ on a probability space $(\Omega, \mathcal{F}, \mathrm{I\!P})$ is given. A *martingale* is a cadlag, adapted stochastic process m which is integrable and satisfies the martingale property:

$$\mathrm{I\!E}\left[\,m(t) \mid \mathcal{F}_s\right] = m(s), \quad s \leq t. \tag{2.1}$$

That is, the increment of the stochastic process $m(t)$ over an arbitrary time interval $(t, t+h]$ given the past has zero expectation. A counting process n is a stochastic process which can be thought of as registering the occurrences in time of a number of discrete events. More formally, a *counting process* is an adapted cadlag process, zero at time zero, with piecewise constant and non-decreasing paths, having jumps of size one only. We say that n has *intensity process* λ, if λ is a predictable process and

$$m(t) := n(t) - \int_0^t \lambda(s)\,\mathrm{d}s \tag{2.2}$$

satisfies the martingale property (2.1). The integral in the right-hand side of (2.2) is often referred to as the *cumulative intensity process* or *compensator* of n. We can consider a martingale as being a pure noise process. The systematic part of a counting process is its compensator, a smoothly varying and predictable process, which subtracted from the counting process leaves unpredictable zero-mean noise. Though m is pure noise, m^2 has a tendency to increase over time. The systematic component (compensator) of m^2 is called the *predictable variation process* of m and denoted by $<m>$. More generally, for martingales m_1 and m_2 the *predictable covariation process* $<m_1, m_2>$ is defined as the unique finite variation cadlag predictable process such that $m_1 m_2 - <m_1, m_2>$ is a martingale, zero at time zero. If h_1 and h_2 are predictable processes, then $\int h_1 \mathrm{d}m_1$ and $\int h_2 \mathrm{d}m_2$ are martingales and $<\int h_1 \mathrm{d}m_1, \int h_2 \mathrm{d}m_2> = \int h_1 h_2 \mathrm{d}<m_1, m_2>$.

Martingales have been studied intensively during the past few decades and a lot of nice mathematical properties have been derived by now. Some very important martingale results are Kurtz' theorems, Lenglart's inequality and the Martingale Central Limit Theorem (MCLT), which can be seen as analogues of the law of large numbers and the usual Central Limit Theorem in the classical i.i.d. case. These results, which will be essential in the proofs of in probability and weak convergence for the non-i.i.d. case (see Chapter 3), are stated explicitly in the next Section 2.4. For a more comprehensive treatment of these and other martingale results we refer to Andersen *et al.* (1993).

Remark 2.1 To get a better understanding of these theoretical concepts we close this section by constructing a special class of counting processes: *the order statistics processes.* It will turn out that this class contains most of (but not all) the popular software reliability models investigated so far. Let us consider a sample of N *independent, identically distributed* survival (or failure) times, $S_1,...,S_N$, from a continuous survival function $S(t)$ with hazard rate function $z(t)$. Hence $z(t)=f(t)/(1-F(t))$ where $F(t)$ is the cumulative distribution function and $f(t)$ the density of the S_i. Typically in survival analysis problems, complete observation of $S_1,...,S_N$ is not possible. In our situation one observes only those S_i that occur in a fixed time interval $[0,\tau]$. We therefore define the *counting process* $n(t)$ for $t \in [0,\tau]$ as

$$n(t) := \# \{ i : S_i \le t \} = \sum_{i=1}^{N} I \{ S_i \le t \},$$

where $I\{.\}$ denotes the indicator function. Thus the stochastic process $n(t)$ is a nondecreasing integer valued function of time with jumps of size one only; it is right continuous and $n(0)=0$. Furthermore we define the stochastic process $Y(t)$, $t \in [0,\tau]$, by

$$Y(t) := \# \{ i : S_i \ge t \} = \sum_{i=1}^{N} I \{ S_i \ge t \} = N - n(t-).$$

Hence $Y(t)$ is the *number at risk* just before time t, or the *size of the risk set.* We define the *intensity process* $\lambda(t)$, the rate at which the counting process $n(t)$ jumps, as:

$$\lambda(t) := Y(t)\, z(t) = \left[N - n(t-) \right] z(t). \tag{2.3}$$

Note that the intensity process λ is random, through dependence on the past of the stochastic process n. Given $\mathcal{F}_{t-} := \sigma \{ n(s),\ s < t \}$, the strict past of n, however, λ is a *predictable* process: that is to say, given $\mathcal{F}_{t-}$ we know $\lambda(t)$ already, but not yet $n(t)$ for instance. It is not difficult to check that in this case the process

$$m(t) := n(t) - \int_0^t \lambda(s)\, \mathrm{d}s$$

indeed satisfies the martingale property (2.1). Counting process models, counting the occurrences of i.i.d. events, and hence having an underlying intensity of the form (2.3), we will call *$N-n$ linear* or *$N-n$ homogeneous.* □

2.4 Asymptotics and limit theory

A major part of this thesis will deal with the study of asymptotic properties of estimators (Chapter 3-5) or with the derivation of the limit-distribution of certain test statistics (Chapter 6). Therefore we should make clear how we treat asymptotics; especially as our approach is rather unorthodox. A novel aspect of our approach, namely, is the fact that—in order to treat asymptotic theory—instead of increasing the time variable or the number of data as is usually the case, we (conceptually) increase one of the model parameters itself.

Recall the formulation of the mathematical problem as sketched in Section 2.3. On a fixed time-interval $\mathcal{T} = [0,\tau]$, we are observing a counting process n with underlying intensity process λ. In the sequel we always assume that this stochastic intensity function, depending on the past of n, is a member of some parametric family:

$$\lambda(t) := \lambda\Big[\, t,\ \theta,\ n(t-)\Big], \quad \theta \in \Theta \subseteq \mathbb{R}^k.$$

We assume the true parameter-value θ_0 is contained in Θ. In all typical cases $\theta_0 = (N_0, \psi_0)$, where N_0, the parameter of most interest, represents the scale or size of the problem (sometimes $N_0 = n(\infty)$), while ψ_0 is a nuisance vector parameter. To apply asymptotics it obviously would not make sense to let τ, the stopping time, grow to infinity. In the long run all failure times will have occurred before time τ and the estimate of the total number of faults N will trivially be equal to the true number of faults. It makes more sense to (conceptually) increase the number of faults N, itself. As we are particularly interested in parameter estimation when N is large, we consider the reparametrisation

$$N = \nu\gamma, \quad \nu \to \infty. \tag{2.4}$$

In (2.4), ν denotes the known scale of the problem (size of the population or some complexity measure of the software program), which we allow to go to infinity. Furthermore, γ represents the unknown proportion coefficient (of ill people or software bugs), which we are going to estimate. In order to make the parametrisation (2.4) profitable we have to put a constraint on λ. $N-n$ linearity (i.e. (2.3)) would be sufficient but is too strong. We require only that the intensity function λ is simultaneously linear in both N and n, that is:

$$\lambda\Big[\, t\,,\ (N,\psi),\ n(t-)\Big] = \alpha^{-1}\,\lambda\Big[\, t,\ (\alpha N,\psi),\ \alpha n(t-)\Big] \quad \alpha > 0. \tag{2.5}$$

Models with intensity function satisfying (2.5) will be called (N,n) *linear* or $(N.n)$ *homogeneous.* We now consider a sequence of counting processes n_ν with increasing ν and underlying (N,n) linear intensities λ_ν and define for $t \in [0,\tau]$

$$x_\nu(t) := \nu^{-1}\, n_\nu(t).$$

The idea behind the transformation (2.4) and (N,n) linearity (2.5) is that $\nu^{-1}\lambda_\nu$ only depends on ν via x_ν. We therefore define

$$\beta\Big[\, t,\ (\gamma,\psi),\ x_\nu(t-)\Big] := \nu^{-1}\,\lambda_\nu\Big[\, t\,;\ (\nu\gamma,\psi),\ \nu x_\nu(t-)\Big].$$

It will turn out to be of great importance that under weak smoothness conditions on β, that are readily satisfied for the models most used in practice, the stochastic process $x_\nu(t)$ will converge in probability uniformly on $[0,\tau]$ to a deterministic function $x_0(t)$, which is the solution of the integral equation

$$x(t) = \int_0^t \beta\left[\, t,\ (\gamma,\psi),\ x(s-)\right] \mathrm{d}s. \tag{2.6}$$

This will follow directly from the next theorem:

Theorem 2.1 *Let $\beta(t;x)$ be a non-anticipating and non-negative function of $t \in \mathcal{T}$ and $x \in D(\mathcal{T})$. Non-anticipating means that β depends only on $x\,|_{[0,t)}$, the past of the stochastic process x up to but not including time t. We assume that*

$$\sup_{s \le t} \beta(t,\ x(s-)) \le C_1 + C_2 \sup_{s<t} |x(s)|,$$

$$\sup_{s \le t} |\beta(t,\ x(s-)) - \beta(t,\ y(s-))| \le C_3 \sup_{s<t} |x(s) - y(s)|,$$

for all $x, y \in D(\mathcal{T})$, and for certain constants C_1, C_2 and C_3. Let $a_\nu \to \infty$ be a sequence of positive constants. Let n_ν be a counting process with underlying intensity process $\lambda_\nu(t) = a_\nu \beta(t,\ a_\nu^{-1} n_\nu(t-))$, for all $t \in \mathcal{T}$. Finally, let x_0 be the unique solution of the integral equation

$$x(t) = \int_0^t \beta(\, t,\ x(s-)\,)\ \mathrm{d}s.$$

Then we have for all $t \in \mathcal{T}$ as $\nu \to \infty$:

$$\sup_{s \le t} |a_\nu^{-1} n_\nu(t) - x_0(t)| \xrightarrow{P} 0.$$

□

This theorem, also known as Kurtz' first theorem, is just the law of large numbers for counting processes. Kurtz' second theorem states that under slightly stronger conditions a central limit theorem result can be obtained, i.e.: $\sqrt{\nu}\,(n_\nu/\nu - x_0) \to Z$ in distribution where Z is a Gaussian process. Both results can be found in Kurtz (1983). One of our main goals in later chapters will be to find estimators for the model parameters and to derive asymptotic properties such as consistency and asymptotic normality. An estimator θ_ν for $\theta_{\theta-}$ is said to be *consistent* if $\theta_\nu \to \theta_0$ in probability and *asymptotically normal* if $\sqrt{\nu}\,(\theta_\nu - \theta_0) \to \mathcal{N}(0,\Sigma)$ in distribution. When deriving these kind of properties in a non-i.i.d. situation we will need the following important martingale results:

Theorem 2.2 *Let m be a local square integrable martingale. Then for any $\eta > 0$ and $\delta > 0$ we have:*

$$\mathrm{IP}\left[\sup_{t\in\mathcal{T}}|m|\geq\eta\right]\leq\frac{\delta}{\eta^2}+\mathrm{IP}\left[<m>(\tau)\geq\delta\right].$$

□

Theorem 2.3 *Consider a sequence n_ν of counting processes with intensity process λ_ν and a sequence of predictable processes H_ν. Define for $t\in\mathcal{T}$,*

$$Z_\nu(t) := \int_0^t H_\nu(s)\left[\mathrm{d}n_\nu(s)-\lambda_\nu(s)\mathrm{d}s\right].$$

Suppose as $\nu\rightarrow\infty$ for all $t\in\mathcal{T}$:

$$<Z_\nu>(t)\overset{P}{\rightarrow}G(t), \qquad (2.7)$$

where G is a continuous function, and suppose that for all $\varepsilon>0$ as $\nu\rightarrow\infty$:

$$\int_0^\tau H_\nu^2(t)\,\lambda_\nu(t)\,I\,\{\,|\,H_\nu(t)\,|\,>\varepsilon\,\}\,\mathrm{d}t\overset{P}{\rightarrow}0. \qquad (2.8)$$

Then Z_ν converges in distribution to Z in the space $D(\mathcal{T})$, where Z is a Gaussian martingale with variance function G and $Z(0)$=0. □

Theorem 2.2, Lenglart's inequality, tells us that we can bound the probability of a large value of m anywhere in the whole time-interval $\mathcal{T}$ in terms just of the probability of a large value of $<m>$ in the endpoint τ. One says that m is dominated by $<m>$. See Lenglart (1977). Theorem 2.3 is a special case of the martingale central limit theorem (MCLT), saying that two conditions are required for a local square integrable martingale to be approximately Gaussian. Condition (2.7) states that its predictable variation process converges in probability to a deterministic function; condition (2.8) states that the jumps of Z_ν become small as $\nu\rightarrow\infty$. For more general formulations of the MCLT and proofs we refer to Rebolledo (1980) or Helland (1982).

2.5 Maximum likelihood estimation

We observe the counting process $n(t)$ on $[0,\tau]$ with underlying (N,n) linear parametric intensity process

$$\lambda(t) = \lambda\left[t,\ (N,\psi),\ n(t-)\right]. \qquad (2.9)$$

The question is now, of course, how to find estimators for N and ψ. We will use the method of Maximum Likelihood Estimation (MLE) for this purpose. Using the fact that $\lambda(t)\mathrm{d}t$ represents the conditional probability given the strict past that the counting process $n(t)$ jumps in the interval $[t,t+\mathrm{d}t]$, we can write for the likelihood:

$$L_\tau\,(N,\psi) \;\propto \prod_{0<t<\tau} \Big[(\,\lambda(t)\mathrm{d}t\,)^{\mathrm{d}n(t)}\,(\,1-\lambda(t)\,\mathrm{d}t\,)^{1-\mathrm{d}n(t)} \Big]$$

$$\propto \;\exp\left[\int_0^\tau \log\lambda(t)\,\mathrm{d}n(t) - \int_0^\tau \lambda(t)\,\mathrm{d}t\right]. \tag{2.10}$$

For a standard definition of the product integral in the upper expression of (2.10) we refer to Gill & Johansen (1990). The lower expression in (2.10) is also known as Jacod's formula (Andersen *et al.* (1993)).

Remark 2.2 Let us again consider the special class of order statistics processes described in Remark 2.1. Thus, given is a population consisting of an unknown number N of failure times S_i. We now furthermore assume that it is given that these failure times are taken from some continuous, parametric distribution function, say $F = F_\psi$. We only observe those failure times S_i, which take a value in $[0,\tau]$. That is, we observe only the order statistics $T_i := S_{(i)}$, $i=1...n$, where n is random, and know that $S_{(n+1)} > \tau$. Equivalently stated in terms of counting processes, we observe the counting process $n(t)$ on $[0,\tau]$ with underlying intensity process

$$\lambda(t) \;=\; \Big[N - n(t-) \Big]\; z(t\,;\psi), \tag{2.11}$$

where $z(t,\psi)=f_\psi(t)/(1-F_\psi(t))$. Again the problem is, how to estimate N and ψ. Of course, we could again use Jacod's expression for the likelihood, but we could also follow the classical approach. Conditioned on the fact that the failure times S_i are i.i.d. F_ψ, the total number of failures in $[0,\tau]$ namely is binomially distributed with parameters N and $F_\psi(\tau)$. Furthermore, conditioned on the fact that S_i is observed, it has a truncated distribution $F_\psi(t)/F_\psi(\tau)$. This leads to the following alternative expression for the likelihood function:

$$L_\tau\,(N,\psi) \;\propto\; \binom{N}{n} F_\psi(\tau)^n\,(\,1-F_\psi(\tau)\,)^{N-n}\; n\,!\,\prod_{i=1}^{n} \frac{f_\psi(T_i)}{F_\psi(\tau)}, \tag{2.12}$$

where the extra factor $n\,!$ in front of the product of truncated densities in (2.12) is explained by the fact that the T_i are the order statistics of the S_i. □

Maximisation of expressions (2.10) (or (2.12)) is usually done by setting partial derivatives of the log-likelihood to zero and solving the the resulting system of highly non-linear likelihood (or score) equations:

$$\frac{\partial}{\partial N}\log L_\tau\,(N,\psi) \;=\; 0, \tag{2.13}$$

$$\frac{\partial}{\partial \psi}\log L_\tau\,(N,\psi) \;=\; 0. \tag{2.14}$$

We have assumed in (2.13) that the model is also meaningful for non-integer N. The direct algebraic solution of the system of non-linear equations (2.13)-(2.14) is usually impossible. The best we can realistically hope for is to solve these equations for a subset of the parameters in terms of the remaining parameters. The remaining parameters are then estimated using numerical methods. The *Newton-Raphson procedure* (Carnahan & Wilkes (1973)) is based on first order Taylor series expansions and has two attractive features. First, if the method converges, it will do so very fast (quadratically). Secondly, convergence is assured if the initial estimate is close enough to a root of the system. This also represents the main drawback of the method: for some initial estimates the method will diverge. This problem becomes severe as the region of feasible values is infinite and high dimensional. Other iterative methods using gradient information (Fletcher & Powell (1963), Fletcher & Reeves (1964)) often also do not display suitable convergence in software reliability practice. In fact, investigations (Musa *et al.* (1987)) indicated that searching schemes incorporating gradient information are not particularly well suited for the problem at hand. The *downhill simplex method* (Nelder & Mead (1965)) does not use any gradient information; it uses only function evaluations (no derivatives). It is very robust and nearly always converges to at least a local minimum (of $-\log L_\tau(N,\psi)$). The main drawback here is that the method lacks any form of acceleration, per se, and therefore tends to be slow. Musa *et al.* (1987) suggest to combine the methods of Newton-Raphson and Nelder-Mead. The idea is that after a limited number of Nelder-Mead iterations the estimate of the solution is close enough to the roots of the system of equations for the fast Newton-Raphson method to converge. Some care should be exercised that the procedure did not accidentally locate a minimum or a saddle-point. More seriously, we should make sure that a global maximum and not a local one has been found. The latter concern can be a problem when using a small sample (of failure times) to estimate multiple parameters simultaneously. Here it is a good idea to have the procedure pick several different starting values and use the estimate with the largest value of $\log L_\tau\ (N,\psi)$. Once the sample size is reasonably large, multiple solutions to the system of likelihood equations (2.13)-(2.14) are rare in typical software reliability models (Moek (1984), Musa *et al.* (1987)), but not impossible (Barendregt & Van Pul (1991)).

2.6 Some software reliability models revisited

In this section we discuss the software reliability models, briefly described in Section 1.5, in more detail. In all examples we assume that software failures are observed during a fixed time-interval $[0,\tau]$ only. We mean by T_i the failure time of the i-th occurring failure, while $t_i=T_i-T_{i-1}$ denotes the interfailure time; that is the time between the $(i-1)$-th and i-th failure. The unknown number of faults initially present in the software is denoted by N_0.

Example 2.1 *The Jelinski-Moranda model (JM).* In the JM-model introduced by Jelinski & Moranda (1972) the failure rate of the program is at any time proportional to the number of remaining faults and the removal of each fault makes the same contribution to the the decay in failure rate. So in terms of counting processes we can write:

$$\lambda^{\mathrm{JM}}(t) = \phi_0\left[N_0 - n(t-)\right],$$

where $\lambda(t)$ denotes the failure rate at time t and where $n(t-)$ denotes the cumulative number of faults detected up to but not including t. The interfailure times t_i are independent and exponentially distributed with parameter $\lambda_i = \phi_0(N_0-i+1)$. Defining

$$\beta^{JM}(\,t\,;\gamma,\phi;x\,) := \phi\left[\gamma-x(t-)\right],$$

it is easy to check that the JM-model is (N,n) linear, that is has a failure intensity λ of the special form (2.4). Solving the integral equation (2.6) yields:

$$x_0^{JM}(t) := \gamma\left[1-e^{-\phi t}\right].$$

In fact, the JM intensity is even $N-n$ linear (of the special form (2.3)) with parametric hazard

$$z^{JM}(t,\phi) := \phi,$$

associated to the exponential distribution. Hence in this case the failure times T_i can be considered as the order statistics of independent and identically distributed exponentials with parameter ϕ. Using (2.10) we can write down the log-likelihood for the JM model:

$$\log L(N,\phi) = \sum_{i=1}^{n(\tau)} \log\phi(N-i+1) - \sum_{i=1}^{n(\tau)+1} \phi(N-i+1)t_i,$$

hence the likelihood equations become

$$\frac{\partial}{\partial N}\log L(N,\phi) = \sum_{i=1}^{n(\tau)} \frac{1}{N-i+1} - \phi\tau = 0; \tag{2.15}$$

$$\frac{\partial}{\partial\phi}\log L(N,\phi) = \frac{n(\tau)}{\phi} - \sum_{i=1}^{n(\tau)+1}(N-i+1)t_i = 0. \tag{2.16}$$

It was shown by Moek (1983) that this system of equations will have a unique solution $(\hat{N},\hat{\phi})$ if and only if the data satisfy:

$$\frac{1}{\tau}\sum_{i=1}^{n(\tau)+1}(i-1)\,t_i > \frac{n(\tau)-1}{2}.$$

Moek's criterion will be satisfied with probability one as N_0 grows larger, if the model is true. □

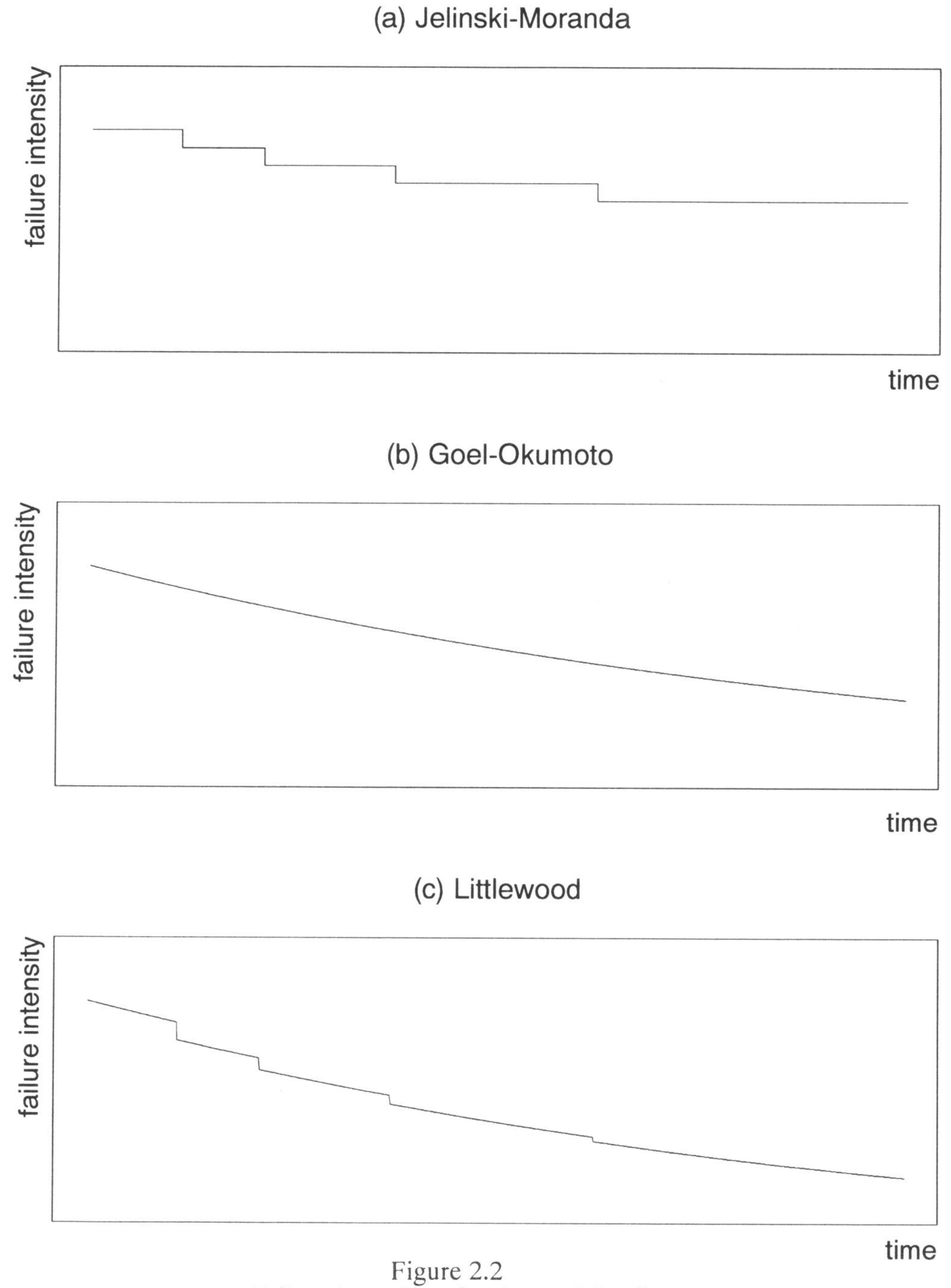


Figure 2.2
Failure intensities for the models of
(a) Jelinski-Moranda, (b) Goel-Okumoto and (c) Littlewood.

Example 2.2 *The Goel-Okumoto model (GO).* In the GO model, suggested by Goel & Okumoto (1979), the failures occur according to a non-homogeneous Poisson-process with failure rate

$$\lambda^{\mathrm{GO}}(t) := N_0 \phi_0 \mathrm{e}^{-\phi_0 t}.$$

Notice that λ does not depend on n; it is a simple deterministic function of time. One can check that the expected number of failures in $[0,\infty)$ equals

$$\mathrm{IE}\left[\int_0^\infty \lambda^{\mathrm{GO}}(s)\,\mathrm{d}s\right] = N_0.$$

Thus we have N_0 faults or sources of failures, each producing failures at an exponentially decreasing rate. The GO model is obviously not $N-n$ linear, but with

$$\beta^{\mathrm{GO}}(\,t,\ \gamma,\phi;x\,) := \gamma\,\phi\,\mathrm{e}^{-\phi t},$$

the GO model satisfies (2.5), having the same deterministic solution x_0 of (2.6) as in the JM case. This means that the JM and GO model are asymptotically equivalent (indistinguishable). In fact, the models are indistinguishable in a stronger sense, since the distribution of the process n in the GO model, conditional on $n(\infty)=k$, is the same as the distribution of the process n in the JM model with $N_0=k$. This means that on the basis of one realisation you cannot distinguish between the models at all. Note that in the GO model the failure rate never becomes zero. This is supposed to reflect the fact that a detected error may or may not be removed and may cause additional errors. However, the exponential hazard rate is completely arbitrary and we feel that the JM model is much more realistic than the GO model. Perhaps the GO model should be considered as an easily analysable approximation to the JM model. □

Example 2.3 *The Littlewood model (L).* The main difference in the L model, introduced by Littlewood (1980), is the fact that each fault does not make the same contribution to the failure rate λ. Littlewood's argument for that is that larger faults will produce failures earlier than smaller ones. He treats the occurrence rate ϕ_i of an individual bug as a stochastic variable. Littlewood's model is an empirical Bayesian model and he himself suggests a gamma distribution $\Gamma(a_0,b_0)$ for the a-priori probability distribution of the ϕ_i. In Section 3.4 it will be derived that the failure rate of the program at time t is then given by

$$\lambda^{\mathrm{L}}(t) := \frac{a_0}{b_0+t}\left[N_0 - n(t-)\right].$$

So as in the JM model, λ depends on the past of the counting process n. Again the failure intensity is (N,n) linear and with

$$\beta^{\mathrm{L}}(\,t\,;\gamma,a,b\,;x\,) := \frac{a}{b+t}\left[\gamma - x(t-)\right]$$

the solution of the integral equation (2.6) becomes

$$x_0^L := \gamma\left[1-(1-t/b)^{-a}\right].$$

Moreover, λ^L is of the special form (2.3), with hazard $z(t;a,b)=a/(b+t)$ and the associated distribution is the Pareto-distribution (mixture of exponentials). We will derive this (and more) in Chapter 5. □

These models will serve as standard examples throughout this thesis. In Chapter 4 simulations according to the JM model are analysed. In Chapter 5 several approaches for the parameter estimation in the L model are investigated. Finally, a new ((N,n) linear, but not $N-n$ linear) model, named the *Poisson Growth & Imperfect Repair & Poisson Growth model (PGIR),* suggested by Van Pul (1991b), is discussed in Chapter 7.

2.7 Overview

In this section we give an overview of the theoretical results that will be presented in Part II of this thesis. In Chapter 3 a rather general class of parametric intensity functions is considered. Following the lines of Borgan (1984) sufficient (but weak) conditions are derived under which some important asymptotic properties (i.e. consistency, asymptotic normality and efficiency) of the maximum likelihood estimators can be proved.

In Chapter 4 we prove that for this general class of intensity functions the asymptotic bootstrap works, i.e. is asymptotically consistent. Furthermore we investigate how well the maximum likelihood estimators behave in practice using simulated data according to the Jelinski-Moranda model and we compare the coverage percentage of confidence intervals constructed with the asymptotic normal and the Wilks test statistic and one using the bootstrap method.

In Chapter 5 we discuss an important statistical problem in parameter estimation, namely that of the possibility of multiple solutions to the system of likelihood equations. Only one of them will be consistent, however. For Littlewood's model we present three solutions to this problem. In a first statistical approach is shown how an initial estimator, that is $\sqrt{n}$-consistent, can be found. We describe an algorithm that produces an asymptotically efficient estimator. In a second, more analytical approach to the problem, we construct an estimator for the maximiser of the likelihood by exploiting its mathematical properties. Moreover, we prove that the global maximiser of the likelihood (in case of the Littlewood model) is consistent using an adapted version of Wald's (1949) classical theory. Inspired by this proof we found a third method of constructing useful estimators by investigating solutions of a slightly modified system of score equations.

Another topic of investigation will be the study of goodness of fit tests. In Chapter 6 we analyse the limit distribution of the Kolmogorov-Smirnov goodness of fit test statistic following the martingale approach suggested by Khmaladze (1981). Only for the class of (inhomogeneous) Poisson processes we could prove that this test statistic is asymptotically distributed as the supremum of a Brownian bridge on [0,1]. For slightly different but less elegant variants of the classical Kolmogorov-Smirnov test statistic we could derive the same limit distribution for processes with a more general intensity.

Finally, of course, one of our ultimate goals will be the study of more realistic models. In Chapter 7 we present the so-called Poisson Growth & Imperfect Repair model (Van Pul (1991b)). We combined the modelling of imperfect repair and software growth in a natural way. Furthermore to a certain extent the model will account for dependencies between faults. The model has attractive statistical properties besides.

Part II

Theory

Chapter 3

Asymptotic Properties*

3.1 Introduction

Computer systems have become more and more important in modern society. The problem of estimating the reliability of computer software has therefore received a great deal of attention over the last two decades. For this purpose a considerable number of models has been proposed. We refer to Musa *et al.* (1987) for a complete overview of the most common software reliability models. Each of these statistical models, based on certain assumptions, is a simplification of reality which we want to describe or understand better. The development of so many different models, which are all supposed to describe the same thing — the evolution of the failure behaviour of a piece of software undergoing debugging — is largely due to a lack of agreement among modellers about how the human mind creates imperfect computer programs. When one wants to predict the reliability of computer software on the basis of past failure data, however, one needs more than just a software reliability model. The model parameter inference procedure and the incorporation of the results in prediction are also very important.

Many important software reliability models can be formulated in terms of counting processes, counting the number of failure occurrences. In this chapter we study some asymptotic properties of the maximum likelihood estimation procedure for parametric counting process models. For a general class of counting processes, we derive conditions

* This chapter is based on CWI-report BS-9029 and on the paper "Asymptotic properties of a class of statistical models in software reliability" by Mark C. van Pul, which appeared in the *Scandinavian Journal of Statistics* (1992), **19,** pp. 235-253.

on the intensity function that are sufficient for these asymptotic properties to hold. We show that the intensity functions of some fairly well-known software reliability models, the models of Jelinski-Moranda and Littlewood, satisfy these conditions. The software models analyzed are perhaps not the most realistic models for computer systems; however, they represent a natural starting point for such a study. A novel aspect of our approach is the fact that — in order to treat asymptotic theory — instead of increasing the time variable or the number of data as is usually the case, we (conceptually) increase one of the model parameters itself. To illustrate the problem and to motivate our concepts, we present here one of the oldest and most elementary software reliability models, that of Jelinski-Moranda (1972), as an example.

Example 3.1 *The Jelinski-Moranda model.* A computer program has been executed during a specified exposure period and the interfailure times are observed. The repairing of a fault takes place immediately after a failure is produced and no new faults are introduced with probability one. Let N be the unknown number of faults initially present in the software. Let the exposure period be $[0,\tau]$ and let $n(t)$, $t \in [0,\tau]$, denote the number of faults detected up to time t. Define $T_0 := 0$ and let T_i, $i=1,2,...,n(\tau)$, be the failure time of the i-th occurring failure, while $t_i := T_i - T_{i-1}$, $i=1,2,...,n(\tau)$, denotes the interfailure time, that is the time between the i-th and the $(i-1)$-th occurring failure. Finally define $t_{n(\tau)+1} := \tau - T_{n(\tau)}$. In the Jelinski-Moranda model, introduced in 1972 and generalised a few years later by Musa (1975), the failure rate of the program at any given time is proportional to the number of remaining faults and each fault still present makes the same contribution to the failure rate. So if $(i-1)$ faults have already been detected, the failure rate for the i-th occurring failure, λ_i, becomes

$$\lambda_i = \phi_0 \left[N_0 - (i-1) \right], \tag{3.1}$$

where ϕ_0 is the true failure rate per fault (the occurrence rate) and N_0 is the true number of faults initially present in the software. In terms of counting processes we can write

$$\lambda^{\mathrm{JM}}(t) = \phi_0 \left[N_0 - n(t-) \right], \; t \in [0,\tau], \tag{3.2}$$

where $\lambda(t)$, $t \in [0,\tau]$ denotes the failure rate at time t. The interfailure times $t_i, i=1,...,n(\tau)$, are independent and exponentially distributed with parameter λ_i given by (3.1). By using the information obtained from the test experiment, one can estimate the the parameters of the underlying model. Maximum likelihood estimation is mostly used for this purpose. Notice that the estimation problem described above is equivalent to that of observing the order statistics $S_{(i)}, i=1...n(\tau)$, in $[0,\tau]$, of an unknown number, N, of exponentials with parameter ϕ. Now let us consider how we treat asymptotic behaviour. It does not make sense to let τ, the stopping time, grow to infinity. In the long run the estimate of the total number of faults will trivially be equal to the true number of faults. It makes more sense to (conceptually) increase the number of faults in the program. The idea is that then asymptotics should be relevant to the practical situation in which N_0 is large and $n(\tau)/N_0$ not close to zero or one. □

In the next section we give some definitions, notation and background. Here we also state more precisely how asymptotic theory is applied. In Section 3.3 (weak) sufficient conditions are given under which we have consistency, asymptotic normality and efficiency of the maximum likelihood estimators (MLE) and local asymptotic normality (LAN) of the model. In the fourth section we discuss an application of the results in software reliability. Two numerical examples based on both real and simulated data are presented in Section 3.5. Finally, in the sixth section a few remarks are given concerning the possibility of weakening some of the conditions. We mention some results from recent investigations, as well as some plans for the future.

3.2 Some definitions, notation and background

[In Section 3.2 we give a brief summary of the relevant concepts, that are more thoroughly discussed in Sections 2.3-2.5.] A *counting process* $n(t)$ is a stochastic process that counts the number of certain events (for instance software failures) up to time t. Thus $n(t)$ is a non-decreasing integer-valued function of time with jumps of size one only; it is right-continuous and $n(0)=0$. A *martingale* $m(t)$ is a stochastic process with the property that the increment over a time-interval $(t,t+h]$ given the past has zero expectation. The past here consists solely of the minimal (or self-exciting) history of the counting process $n(t)$. In regular cases, a counting process $n(t)$ is accompanied by an *intensity process* $\lambda(t)$. It is interpreted heuristically as the probability rate that $n(t)$ jumps in a small time interval $[t,t+\mathrm{d}t]$ at t, conditioned on the past. A more formal definition states that $\lambda(t)$ is the intensity of $n(t)$ if it is *predictable* (intuitively: non-stochastic given the strict past) and

$$m(t) := n(t) - \int_0^t \lambda(s)\mathrm{d}s$$

is a martingale. Formal definitions of these concepts can be found in Andersen *et al.* (1993).

Let a counting process $n(t)$ be given. Jumps of the counting process $n(t)$ are observed only during a specific time interval $[0,\tau]$. We assume that the intensity function associated with the counting process exists and is a member of some specified parametric family, that is:

$$\lambda(t) := \lambda(t;N,\psi),\ t\in[0,\tau],\ N\in\mathrm{IN},$$

for an integer p. Let N_0 and ψ_0 be the true parameter values. Typically the parameter N_0 represents the scale or size of the problem (sometimes $N_0=n(\infty)$), while ψ_0 is a nuisance vector parameter. We are interested in the estimation of N_0 and ψ_0 as $N_0\rightarrow\infty$. We assume that the model is also meaningful for non-integer N. For instance the intensity function (3.2) of the Jelinski-Moranda model can be generalised to

$$\lambda^{\mathrm{JM}}(t) = \phi\left[N - n(t-)\right] I\,\{\,n(t) < N\,\},\quad t\in[0,\tau],$$

where $I\{.\}$ denotes the indicator function. As we are particularly interested in parameter estimation when N_0 is large, we introduce a series of counting processes $n_\nu(t)$, $t\in[0,\tau], \nu=1,2,\ldots$ and let N_0 conceptually increase. Let $N = N_\nu \rightarrow \infty$ for $\nu \rightarrow \infty$.

By the reparametrisation

$$N_\nu \;=\; \nu\gamma_\nu$$

with a dummy variable γ_ν, we can denote the associated intensity functions by

$$\lambda_\nu(t\,;\gamma,\psi) := \lambda(t\,;\nu\gamma,\psi),\; t\in[0,\tau],\;\; \gamma\in\mathrm{IR}^+,$$

Now we consider the estimation of γ and ψ as $\nu\to\infty$. We treat ν as known and γ as unknown. If the real-life situation has $\nu=N_0$, then $\gamma=\gamma_0=1$ and $\psi=\psi_0$.

It is rather unorthodox to increase a model parameter itself, in this case N. This complication is solved by estimating γ. We assume that the maximum likelihood estimators $(\hat\gamma_\nu,\hat\psi_\nu)$ for (γ_0,ψ_0) exist. Typically, $(\hat\gamma_\nu,\hat\psi_\nu)$ is a root of the likelihood equations

$$\frac{\partial}{\partial(\gamma,\psi)}\log L_\nu(\gamma,\psi;\tau) = 0,\quad \nu=1,2,...,$$

where the likelihood function at time t is given by (see Aalen (1978), Andersen *et al.* (1993))

$$L_\nu(\gamma,\psi;t) := \exp\left[\int_0^t \log\lambda_\nu(s\,;\gamma,\psi)\,\mathrm{d}n_\nu(s) - \int_0^t \lambda_\nu(s\,;\gamma,\psi)\,\mathrm{d}s\right].$$

We will take care later to rephrase results on the behaviour of $\hat\gamma_\nu$ as $\nu\to\infty$ in terms of $\hat N$ as $N_0\to\infty$, where $\hat N=\nu\hat\gamma_\nu$. By invariance of the maximum likelihood estimation method the value of ν chosen in actual computations does not influence the value of the result $\hat N$. Also, estimated asymptotic variances etc. for $\hat N$ depend on ν and $\hat\gamma_\nu$ only through $\nu\hat\gamma_\nu$. As we said information obtained for the asymptotic behaviour of $\hat\gamma_\nu$ can be transformed back directly to $\hat N$, the estimator of main interest. More precisely, consistency of $\hat\gamma$ (or $\hat\gamma_\nu \to_p \gamma_0$ as $\nu\to\infty$) implies $\hat N/N_0 \;\to_p\; 1$ as $N_0\to\infty$. Similarly concerning the asymptotic normality:

$$\sqrt{\nu}\left[\hat\gamma_\nu - \gamma_0\right] \to_D \mathcal{N}\left[0,\sigma^2(\gamma_0,\psi_0)\right] \Rightarrow \sqrt{N_0}\left[\frac{\hat N}{N_0}-1\right] \to_D \mathcal{N}\left[0,\frac{\sigma^2(\gamma_0,\psi_0)}{\gamma_0}\right], \quad (3.3)$$

where $\gamma_0=1$ and ν and N_0 respectively tend to infinity. Result (3.3) states that $\hat N$ is asymptotically normally distributed with mean N_0 and variance $N_0\sigma^2(\gamma_0,\psi_0)/\gamma_0$. One will use this result in practice by estimating the variance as $\hat N\sigma^2(\hat\gamma,\hat\psi)/\hat\gamma$. This quantity turns out not to depend on $\hat\gamma$ (see also Remark 3.3 and (3.25) below). Alternatively, if one uses observed Fisher information, one also gets parametrisation-free conclusions, immediately.

Finally, we know from the theory of counting processes that

$$m_\nu(t;\gamma,\psi) := n_\nu(t) - \int_0^t \lambda_\nu(s;\gamma,\psi)\mathrm{d}s, \quad \nu=1,2,...,$$

are local square integrable martingales. We define for $\nu=1,2,...$ the stochastic process $x_\nu(t)$ by

$$x_\nu(t) := \nu^{-1} n_\nu(t), \quad t\in[0,\tau]. \tag{3.4}$$

In some important practical situations, as we shall soon see, this stochastic process converges uniformly on $[0,\tau]$ in probability to a deterministic function $x_0(t)$ as $\nu\to\infty$. In the next section we give (weak) sufficient conditions for intensity functions λ_ν of a certain form under which we have consistency, asymptotic normality and efficiency of the maximum likelihood estimators (MLE) and local asymptotic normality (LAN) of the model.

3.3 Asymptotic properties

We consider a sequence of models $(\lambda_\nu, m_\nu, x_\nu), \nu=1,2,...$ as defined in the previous section. For reasons of notational convenience we take $\theta:=(\gamma,\psi)^T\in\Theta$, $\Theta\subset\mathrm{IR}^p$ for some integer p. In the sequel we assume that the intensity function λ_ν is of the form:

$$\lambda_\nu(t;\theta) = \nu\beta(t,\theta,x_\nu), \tag{3.5}$$

where β: $[0,\tau]\times\Theta\times K \to \mathrm{IR}^+$ is an arbitrary non-negative and non-anticipating function. Non-anticipating means that $\beta(t,\theta,x_\nu)$ only depends on $x_\nu|_{[0,t)}$, the past of the stochastic process x_ν up to but not including time t. In fact, in most practical cases $\beta(t,\theta,x_\nu)$ will depend only on $x_\nu(t-)$. On $K:=D(\,[0,\tau]\,)$, the space of right-continuous functions on $[0,\tau]$ with left limits (so-called cadlag functions), we put the usual supremum norm. The likelihood function $L_\nu(\theta,t)$ now becomes for $\theta\in\Theta$, $t\in[0,\tau]$ and $\nu=1,2,...$:

$$L_\nu(\theta,t) := \exp\left[\int_0^t \log \nu\beta(s,\theta,x_\nu)\mathrm{d}n_\nu(s) - \nu\int_0^t \beta(s,\theta,x_\nu)\mathrm{d}s\right]. \tag{3.6}$$

Furthermore, we define for $\theta\in\Theta$, $t\in[0,\tau]$, $\nu=1,2,...$:

$$C_\nu(\theta,t) := \log L_\nu(\theta,t), \tag{3.7}$$

$$U_{\nu i}(\theta,t) := \frac{\partial}{\partial\theta_i} C_\nu(\theta,t), \tag{3.8}$$

$$I_{\nu ij}(\theta,t) := \frac{\partial^2}{\partial\theta_i\partial\theta_j} C_\nu(\theta,t), \tag{3.9}$$

$$R_{\nu ijk}(\theta,t) := \frac{\partial^3}{\partial\theta_i\partial\theta_j\partial\theta_k} C_\nu(\theta,t). \tag{3.10}$$

Consider the following global conditions:

(G1) For all $x \in K$ and for all $\theta \in \Theta$ the intensity function β satisfies

$$\sup_{t \leq \tau} \beta(t,\theta,x) < \infty.$$

(G2) (Lipschitz continuity) For all $\theta \in \Theta$ there exists a constant L, such that for all $x,y \in K$ and all $t \in [0,\tau]$

$$|\beta(t,\theta,x) - \beta(t,\theta,y)| \leq L \sup_{s \leq t} |x(s) - y(s)|.$$

Under the global conditions (G1)–(G2) the stochastic process $x_\nu(t)$, as defined in (3.4), converges uniformly on $[0,\tau]$ in probability to $x_0(t)$ as $\nu \to \infty$, where $x_0 \in D([0,\tau])$ is the unique solution of

$$x(t) = \int_0^t \beta(s,\theta_0,x)\mathrm{d}s.$$

This was proved by Kurtz (1983).

Next, we consider the following local conditions:

(L1) There exist neighbourhoods Θ_0 and K_0 of θ_0, x_0 respectively, such that the function $\beta(t,\theta,x)$ and its derivatives with respect to θ of the first, second and third order exist, are continuous functions of θ and x, bounded on $[0,\tau] \times \Theta_0 \times K_0$.

(L2) The function $\beta(t,\theta,x)$ is bounded away from zero on $[0,\tau] \times \Theta_0 \times K_0$.

(L3) The matrix $\Sigma = \{\sigma_{ij}(\theta_0)\}$ is positive definite, with for $i,j \in \{1,2,..,p\}$, $\theta \in \Theta_0$:

$$\sigma_{ij}(\theta) := \int_0^\tau \frac{\frac{\partial}{\partial\theta_i}\beta(s,\theta,x_0)\frac{\partial}{\partial\theta_j}\beta(s,\theta,x_0)}{\beta(s,\theta,x_0)} \mathrm{d}s. \tag{3.11}$$

We are now able to formulate the main result.

Theorem 3.1 *Consider a counting process with intensity function $\lambda(t;N,\psi)$, where (N,ψ) denotes an unknown p-dimensional parameter. As in Section 3.2 we can define an associated sequence of experiments by letting $\nu\rightarrow\infty$. Let $\theta_0=(\gamma_0,\psi_0)$ be the true value of the parameter. Assume that for all ν the intensity function $\lambda_\nu(t;\theta)$ in the ν-th experiment is of the form (3.5) for a certain function β satisfying conditions (G1)–(G2) and (L1)–(L3). Then we have:*

(i) *Consistency of ML-estimators: With probability tending to 1, the likelihood equations*

$$\frac{\partial}{\partial\theta}\log L_\nu(\theta,\tau)=0, \quad \nu=1,2,... \tag{3.12}$$

have exactly one consistent solution $\hat{\theta}_\nu$. Moreover this solution provides a local maximum of the likelihood function (3.6).

(ii) *Asymptotic normality of the ML-estimators: Let $\hat{\theta}_\nu$ be the consistent solution of the maximum likelihood equations (3.12), then*

$$\sqrt{\nu}\,(\hat{\theta}_\nu-\theta_0)\rightarrow_{D(\theta_0)}\mathcal{N}(0,\Sigma^{-1}), \quad \nu\rightarrow\infty,$$

where Σ is given by (3.11) and can be estimated consistently from the observed information matrix I_ν, given in (3.9).

(iii) *Local asymptotic normality of the model: With $U_\nu=U_\nu(\theta_0,\tau)$ given by (3.8), we have for all $h\in\mathrm{IR}^p$:*

$$\log\frac{d\mathrm{IP}_{\theta_\nu}}{d\mathrm{IP}_{\theta_0}} - \nu^{-\frac{1}{2}}h^TU_\nu + \frac{1}{2}h^T\Sigma h \rightarrow_{\mathrm{IP}_{\theta_0}} 0, \quad \nu\rightarrow\infty, \tag{3.13}$$

where $\theta_\nu=\theta_0+\nu^{-\frac{1}{2}}h$ and $\nu^{-\frac{1}{2}}U_\nu\rightarrow_D\mathcal{N}(0,\Sigma)$.

(iv) *Asymptotic efficiency of the ML-estimators: $\hat{\theta}_\nu$ is asymptotically efficient in the sense that it is regular and the limit distribution for any other regular estimator $\tilde{\theta}_\nu$ for θ_0 satisfies*

$$\sqrt{\nu}\,(\tilde{\theta}_\nu-\theta_0)\rightarrow_{D(\theta_0)} Z+Y,$$

where $Z\sim_d\mathcal{N}(0,\Sigma^{-1})$, Z and Y independent. (For a definition of the regularity of an estimator we refer to van der Vaart (1987), Ibragimov & Khas'minskii (1979) or to (3.17) below.)

[In Chapter 4 we will see that (3.13) still holds for sequences $\theta_\nu=\theta_0+\nu^{-1/2}h+o(h)$, that is the model satisfies the strong LAN property (SLAN). Together with the asymptotic normality this implies strong regularity of the ML-estimator.]

Remark 3.1 A nearly immediate consequence of these results about the asymptotic distribution of the ML-estimator $\hat{\theta}_\nu$ is the fact that the Wald test statistic

$$-(\hat{\theta}_\nu-\theta_0)^T I_\nu(\hat{\theta}_\nu,\tau)(\hat{\theta}_\nu-\theta_0),$$

where I_ν is given by (3.9), is asymptotically chi-squared distributed with p degrees of freedom under the simple hypothesis $H_0 : \theta=\theta_0$. With C_ν, U_ν and I_ν given by (3.7)–(3.9) the Rao test (or score) statistic

$$-U_\nu(\theta_0,\tau)^T I_\nu(\theta_0,\tau)^{-1} U_\nu(\theta_0,\tau)$$

and the Wilks test (or likelihood ratio) statistic

$$2\left[C_\nu(\hat{\theta}_\nu,\tau)-C_\nu(\theta_0,\tau)\right] \tag{3.14}$$

have the same asymptotic distribution as the Wald test statistic. Equivalence of these tests can be shown by the arguments of Rao (1973). □

Proof Theorem 3.1 One can easily check that (G1)–(G2) & (L1)–(L3) imply the following more classical looking set of conditions (C1)–(C3) (see Andersen *et al.* (1993)):

(C1) The function β is continuous with respect to θ, and strictly positive.

(C2) There exists a non-negative deterministic function $x_0 \in K$ and neighbourhoods Θ_0, K_0 of θ_0 and x_0 respectively, such that the derivatives of $\beta(t,\theta,x)$ with respect to θ of the first, second and third order exist and are continuous functions of θ, on $[0,\tau]\times\Theta_0\times K_0$. With $x_\nu, \nu=1,2,\ldots$ the stochastic process given in (3.4), $x_0 \in K$ has to satisfy for all $i,j \in \{1,2,\ldots,\mathrm{p}\}$ as $\nu \to \infty$:

$$\int_0^\tau \frac{\frac{\partial}{\partial\theta_i}\beta(s;\theta_0,x_\nu)\frac{\partial}{\partial\theta_j}\beta(s;\theta_0,x_\nu)}{\beta(s;\theta_0,x_\nu)}ds \rightarrow_{\mathrm{IP}} \int_0^\tau \frac{\frac{\partial}{\partial\theta_i}\beta(s;\theta_0,x_0)\frac{\partial}{\partial\theta_j}\beta(s;\theta_0,x_0)}{\beta(s;\theta_0,x_0)}ds < \infty$$

$$\int_0^\tau \left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\beta(s;\theta_0,x_\nu)\right]^2 \beta(s;\theta_0,x_\nu)ds \rightarrow_{\mathrm{IP}} \int_0^\tau \left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\beta(s;\theta_0,x_0)\right]^2 \beta(s;\theta_0,x_0)ds < \infty.$$

(C3) There exist functions G and H and neighbourhoods Θ_0, K_0 of θ_0 and x_0 respectively, such that for all $t \in [0,\tau]$ and $x \in K_0$:

$$\sup_{\theta\in\Theta_0} \left| \frac{\partial^3}{\partial\theta_i\partial\theta_j\partial\theta_k}\beta(t;\theta,x) \right| \leq G(t,x),$$

$$\sup_{\theta\in\Theta_0} \left| \frac{\partial^3}{\partial\theta_i\partial\theta_j\partial\theta_k} \log\beta(t,\theta,x) \right| \le H(t,x),$$

and moreover the functions G and H satisfy as $\nu\to\infty$:

$$\int_0^\tau G(s,x_\nu)\mathrm{d}s \to_{\mathrm{IP}} \int_0^\tau G(s,x_0)\mathrm{d}s < \infty,$$

$$\int_0^\tau H(s,x_\nu)\beta(s,\theta_0,x_\nu)\mathrm{d}s \to_{\mathrm{IP}} \int_0^\tau H(s,x_{0)}\beta(s,\theta_0,x_{0)}\mathrm{d}s < \infty,$$

$$\int_0^\tau H^2(s,x_\nu)\beta(s,\theta_0,x_\nu)\mathrm{d}s \to_{\mathrm{IP}} \int_0^\tau H^2(s,x_0)\beta(s,\theta_0,x_0)\mathrm{d}s < \infty.$$

Although our model (3.5) is not a special case of the multiplicative intensity model considered in Borgan (1984), the rest of the proof of (i) and (ii) of Theorem 3.1, which is given in Van Pul (1990), now follows exactly the lines of Borgan (1984) and is omitted here. Borgan starts with conditions of the type (C1)–(C3) and uses the same standard argumentation as given by Cramér (1946), who derived similar results for the classical case of i.i.d. random variables. Compared with the i.i.d. case the difference is that in the present context Lenglart's inequality is used to establish the convergence in probability results (instead of the law of large numbers in the classical case), while we have to use the martingale central limit theorem to establish the weak convergence result, which in the classical case is proved by the central limit theorem for i.i.d. random variables. [For sake of completeness we have included the proofs of (i) and (ii) of Theorem 3.1 in this thesis.]

(i) Consistency of ML-estimators: By a Taylor series expansion we get for any $\theta\in\Theta$:

$$U_{\nu i}(\theta,\tau) = U_{\nu i}(\theta_0,\tau)+\sum_{j=1}^{p}(\theta_j-\theta_{j0})I_{\nu ij}(\theta_0,\tau) + \frac{1}{2}\sum_{j=1}^{p}\sum_{k=1}^{p}(\theta_j-\theta_{j0})(\theta_k-\theta_{k0})R_{\nu ijk}(\theta_\nu^*,\tau), \quad (3.15)$$

where $\theta_\nu^*=\theta_\nu^*(\theta)$ is on the line segment joining θ and θ_0. We shall show that:

$$\nu^{-1}U_{\nu i}(\theta_0,\tau) \to_{\mathrm{IP}} 0, \qquad (3.16)$$

$$\nu^{-1}I_{\nu ij}(\theta_0,\tau) \to_{\mathrm{IP}} -\sigma_{ij}(\theta_0), \qquad (3.17)$$

$$\lim_{\nu\to\infty} \mathrm{IP}\left(\left|\nu^{-1}R_{\nu ijk}(\theta,\tau)\right| < M \right) = 1, \qquad (3.18)$$

for all $i,j,k\in\{1,2,...,p\}$, all $\theta\in\Theta_0$ and a certain finite constant M, not depending on θ.

From (3.16)-(3.18) the statement (i) in the theorem will follow by a standard argument (see Billingsley (1961), pp. 10-16).

Let us first prove (3.16). From (3.5) we get that (3.8) evaluated at the true parameter θ_0, equals:

$$U_{\nu i}(\theta_0,t) \;=\; \frac{\partial}{\partial\theta_i}\left[\int_0^t \log\nu\beta(s,\theta_0,x_\nu)\mathrm{d}n_\nu(s) - \int_0^t \nu\beta(s,\theta_0,x_\nu)\mathrm{d}s\right]$$

$$= \int_0^t \frac{\partial}{\partial\theta_i}\Big[\log\beta(s,\theta_0,x_\nu)\Big]\mathrm{d}\left[m_n(s,\theta_0)+\int_0^s \nu\beta(u,\theta_0,x_\nu)\mathrm{d}u\right] - \nu\int_0^t \frac{\partial}{\partial\theta_i}\beta(s,\theta_0,x_\nu)\mathrm{d}s$$

$$= \int_0^t \frac{\frac{\partial}{\partial\theta_i}\beta(s,\theta_0,x_\nu)}{\beta(s,\theta_0,x_\nu)}\mathrm{d}m_\nu(s,\theta_0). \tag{3.19}$$

Because β is a non-anticipating function, it follows that $U_{\nu i}(\theta_0,t)$ is a stochastic integral of a predictable process w.r.t. a local martingale and hence a local square integrable martingale. Its variance process equals

$$<U_{\nu i}(\theta_0,\cdot),U_{\nu i}(\theta_0,\cdot)>(t) = \int_0^t \left[\frac{\frac{\partial}{\partial\theta_i}\beta(s,\theta_0,x_\nu)}{\beta(s,\theta_0,x_\nu)}\right]^2 \nu\beta(s,\theta_0,x_\nu)\mathrm{d}s. \tag{3.20}$$

By condition (C2), $\nu^{-1}<U_{\nu i}(\theta_0,\cdot),U_{\nu i}(\theta_0,\cdot)>(\tau)$ converges in probability to some finite quantity as $\nu\to\infty$. Therefore, an application of Lenglart's inequality (see Andersen & Gill (1982)) gives that for all $\delta,\eta>0$ we have

$$\mathrm{IP}\left[\sup_{t\in[0,\tau]}\left|\nu^{-1}U_{\nu i}(\theta_0,t)\right| > \eta\right] \;\le\; \frac{\delta}{\eta^2} + \mathrm{IP}\left[\nu^{-2}<U_{\nu i}(\theta_0,\cdot),U_{\nu i}(\theta_0,\cdot)>(\tau) > \delta\right] \tag{3.21}$$

which proves (3.16).

To prove (3.17), note that by using (3.5):

$$\nu^{-1}I_{\nu ij}(\theta_0,\tau) \;=\; \nu^{-1}\frac{\partial^2}{\partial\theta_i\partial\theta_j}\left[\int_0^\tau \log\nu\beta(s,\theta_0,x_\nu)\mathrm{d}n_\nu(s) - \int_0^\tau \nu\beta(s,\theta_0,x_\nu)\mathrm{d}s\right]$$

$$= \nu^{-1}\int_0^\tau \frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\beta(s,\theta_0,x_\nu)\mathrm{d}\left(m_\nu(s,\theta_0) + \int_0^s \nu\beta(u,\theta_0,x_\nu)\mathrm{d}u \right)$$

$$- \int_0^\tau \frac{\partial^2}{\partial\theta_i\partial\theta_j}\beta(s,\theta_0,x_\nu)\mathrm{d}s$$

$$= \nu^{-1}\int_0^\tau \frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\beta(s,\theta_0,x_\nu)\mathrm{d}m_\nu(s,\theta_0)$$

$$+ \int_0^\tau \left[\beta(s,\theta_0,x_\nu)\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\beta(s,\theta_0,x_\nu) - \frac{\partial^2}{\partial\theta_i\partial\theta_j}\beta(s,\theta_0,x_\nu) \right]\mathrm{d}s$$

$$= \nu^{-1}\int_0^\tau \frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\beta(s,\theta_0,x_\nu)\mathrm{d}m_\nu(s,\theta_0)$$

$$- \int_0^\tau \frac{\frac{\partial}{\partial\theta_i}\beta(s,\theta_0,x_\nu)\frac{\partial}{\partial\theta_j}\beta(s,\theta_0,x_\nu)}{\beta(s,\theta_0,x_\nu)}\mathrm{d}s. \tag{3.22}$$

That the first of the last two terms in (3.22) converges in probability to zero follows by condition (C2) and an application of Lenglart's inequality similar to (3.21). By (G2), (3.17) is an immediate consequence.

Finally to prove (3.18), we note that the first two inequalities in (C3) give for all $i,j,k\in\{1,2,...,p\}$ and all $\theta\in\Theta_0$:

$$\left|\nu^{-1}R_{\nu ijk}(\theta,\tau)\right| = \left|\nu^{-1}\frac{\partial^3}{\partial\theta_i\partial\theta_j\partial\theta_k}\left(\int_0^\tau \log\nu\beta(s,\theta,x_\nu)\mathrm{d}n_\nu(s) - \int_0^\tau \nu\beta(s,\theta,x_\nu)\mathrm{d}s \right)\right|$$

$$\le \nu^{-1}\int_0^\tau \left|\frac{\partial^3}{\partial\theta_i\partial\theta_j\partial\theta_k}\log\beta(s,\theta,x_\nu)\right|\mathrm{d}n_\nu(s) + \int_0^\tau \left|\frac{\partial^3}{\partial\theta_i\partial\theta_j\partial\theta_k}\beta(s,\theta,x_\nu)\right|\mathrm{d}s$$

$$\le \nu^{-1}\int_0^\tau H(t,x_\nu)\mathrm{d}n_\nu(s) + \int_0^\tau G(t,x_\nu)\mathrm{d}s. \tag{3.23}$$

By another application of Lenglart's inequality we get for all $\delta,\eta>0$:

$$\mathrm{IP}\left(\sup_{t\in[0,\tau]} \left| \nu^{-1}\int_0^t H(s,x_\nu)\mathrm{d}n_\nu(s) - \nu^{-1}\int_0^t H(s,x_\nu)\nu\beta(s,\theta_0,x_\nu)\mathrm{d}s \right| > \eta \right)$$

$$\leq \frac{\delta}{\eta^2} + \mathrm{IP}\left(\nu^{-2}\int_0^\tau H^2(s,x_\nu)\nu\beta(s,\theta_0,x_\nu)\mathrm{d}s \right). \tag{3.24}$$

So the last two expressions in (C3) yield

$$\nu^{-1}\int_0^\tau H(s,x_\nu)\mathrm{d}n_\nu(s) \rightarrow \int_0^\tau H(s,x)\beta(s,\theta_0,x)\mathrm{d}s \tag{3.25}$$

in probability as $\nu\rightarrow\infty$. Combining this with the third expression in (C3) and with (3.24), we get (3.18) and hence the consistency of the ML-estimators. □

(ii) Asymptotic normality of the ML-estimators: Let $\hat{\theta}_\nu$ be the consistent solution of the likelihood equations (3.12). Taylor expanding $U_{\nu i}(\theta_\nu,\tau)$ around θ_0 gives:

$$0 = \nu^{-\frac{1}{2}} U_{\nu i}(\hat{\theta}_\nu,\tau)$$

$$= \nu^{-\frac{1}{2}} U_{\nu i}(\theta_0,\tau) + \sum_{j=1}^{p} \nu^{\frac{1}{2}} (\hat{\theta}_{\nu j}-\theta_{0j})\nu^{-1} I_{\nu ij}(\theta_\nu^*,\tau), \tag{3.26}$$

where $\theta_\nu^*=\theta_\nu^*(\hat{\theta}_\nu)$ is on the line segment between $\hat{\theta}_\nu$ and θ_0. Therefore, the statement (ii) in Theorem 1 will follow by an argument in Billingsley (1961), if we can prove that for all $i,j\in\{1,2,\ldots,p\}$ and for any random θ_ν^*, such that $\theta_\nu^*\rightarrow\theta_0$ in probability as $\nu\rightarrow\infty$:

$$\nu^{-\frac{1}{2}} U_\nu(\theta_0,\tau) \rightarrow_D \mathrm{N}(0,\Sigma) \tag{3.27}$$

$$\nu^{-1} I_{\nu ij}(\theta_\nu^*,\tau) \rightarrow_{\mathrm{IP}} -\sigma_{ij}(\theta_0). \tag{3.28}$$

Let us first prove (3.27). By (3.19) and condition (C2)

$$< \nu^{-\frac{1}{2}} U_{\nu i}(\theta_0,\cdot),\ \nu^{-\frac{1}{2}} U_{\nu j}(\theta_0,\cdot) >(\tau)$$

$$= \int_0^\tau \frac{\partial}{\partial\theta_i}\log\beta(s,\theta_0,x_\nu)\frac{\partial}{\partial\theta_j}\log\beta(s,\theta_0,x_\nu)\cdot\beta(s,\theta_0,x_\nu)\mathrm{d}s$$

$$= \int_0^\tau \frac{\frac{\partial}{\partial\theta_i}\beta(s,\theta_0,x_\nu)\frac{\partial}{\partial\theta_j}\beta(s,\theta_0,x_\nu)}{\beta(s,\theta_0,x_\nu)}\mathrm{d}s$$

$$\rightarrow_{\mathrm{IP}} \int_0^\tau \frac{\frac{\partial}{\partial\theta_i}\beta(s,\theta_0,x)\frac{\partial}{\partial\theta_j}\beta(s,\theta_0,x)}{\beta(s,\theta_0,x)}\mathrm{d}s \tag{3.29}$$

as $\nu\rightarrow\infty$ for all $i,j\in\{1,2,...,p\}$. Furthermore condition (C2) and dominated convergence give that for all $j\in\{1,2,...,p\}$ and all $\varepsilon>0$ as $\nu\rightarrow\infty$:

$$\int_0^\tau \left(\frac{\frac{\partial}{\partial\theta_i}\beta(s,\theta_0,x_\nu)}{\nu^{\frac{1}{2}}\beta(s,\theta_0,x_\nu)}\right)^2 \nu\beta(s,\theta_0,x_\nu)\, I\left\{\left|\frac{\frac{\partial}{\partial\theta_i}\beta(s,\theta_0,x_\nu)}{\nu^{\frac{1}{2}}\beta(s,\theta_0,x_\nu)}\right| > \varepsilon\right\}\mathrm{d}s \rightarrow_{\mathrm{IP}} 0 \tag{3.30}$$

in probability as $\nu\rightarrow\infty$. From this (3.27) follows by an application of the martingale central limit theorem (see Andersen & Gill (1982)).

Let us return to (3.28). By a Taylor series expansion we have when $\theta_\nu^*\in\Theta$:

$$\nu^{-1}I_{\nu ij}(\theta_\nu^*,\tau) = \nu^{-1}I_{\nu ij}(\theta_0,\tau)+\nu^{-1}\sum_{k=1}^{p}(\theta_{\nu k}^*-\theta_{0k})R_{\nu ijk}(\bar{\theta}_\nu,\tau), \tag{3.31}$$

where $R_{\nu ijk}(\theta,t)$ is defined in (3.10) and $\bar{\theta}_\nu=\bar{\theta}_\nu(\theta_\nu^*)$ is on the line segment joining θ_ν^* and θ_0. By (3.17) and (3.18), the first term in the right hand side of (3.31) converges in probability to $-\sigma_{ij}(\theta_0)$ as $\nu\rightarrow\infty$, while the second term is bounded in probability by $pM|\theta_\nu^*-\theta_0|_p$ for some finite constant M not depending on θ_ν^*. This proves (3.28) and thus the asymptotic normality of the ML-estimators. □

We will now give proofs of part (iii) and (iv) of Theorem 3.1. It should be noted that Hjort in the discussion of the lecture of Andersen & Borgan (1985) already pointed out that Local Asymptotic Normality in Borgan's model could easily have been shown by him. In Hjort (1986) LAN is proved for the multiplicative model. See also Andersen *et al.* (1993).

(iii) Local asymptotic normality of the model. For sake of convenience, we introduce some more notation. For a function $f:\mathrm{IR}^p\rightarrow\mathrm{IR}$, which is at least three times differentiable, we write:

$$\frac{\partial^3}{\partial x^3} f(x_0) \left(\frac{\partial^3}{\partial x_i \partial x_j \partial x_k} f(x_0) \right)_{1 \le i,j,k \le p},$$

the (three-dimensional) $p \times p \times p$ matrix of third order partial derivatives, evaluated in x_0. Furthermore, for a (three-dimensional) $p \times p \times p$ matrix $Y = (y_{ijk})$, and a p-vector $g = (g_i)$, we define:

$$g^T Y g^{<2>} := \sum_{i=1}^{p} \sum_{j=1}^{p} \sum_{k=1}^{p} g_i g_j g_k y_{ijk}.$$

We define for $h \in \Theta$:

$$\theta_\nu(h) \; := \; \theta_0 + \nu^{-\frac{1}{2}} h, \; \nu = 1, 2, \ldots.$$

For fixed h and ν, using the fact that $\lambda_\nu = \nu\beta$, we have that the log-likelihood ratio for $\theta_\nu(h)$ against θ_0 is given by

$$\begin{aligned} Q_\nu(h) &= \log \frac{\mathrm{dIP}_{\theta_\nu(h)}}{\mathrm{dIP}_{\theta_0}} \\ &= \log \mathrm{dIP}_{\theta_\nu(h)} - \log \mathrm{dIP}_{\theta_0} \\ &= \left[\int_0^\tau \log \lambda_\nu(s, \theta_\nu(h)) \mathrm{d}n_\nu(s) - \int_0^\tau \lambda_\nu(s, \theta_\nu(h)) \mathrm{d}s \right] \\ &\quad - \left[\int_0^\tau \log \lambda_\nu(s, \theta_0) \mathrm{d}n_\nu(s) - \int_0^\tau \lambda_\nu(s, \theta_0) \mathrm{d}s \right] \\ &= C_\nu(\theta_\nu(h), \tau) - C_\nu(\theta_0, \tau), \end{aligned}$$

where C_ν is given by (3.7). Of course $Q_\nu(0) = 0$, and because

$$\frac{\partial}{\partial h} \theta_\nu(h) = \nu^{-\frac{1}{2}},$$

the first, second and third order derivatives of Q_ν with respect to h are:

$$\frac{\partial}{\partial h} Q_\nu(h) = \nu^{-\frac{1}{2}} U_\nu(\theta_\nu(h),\tau),$$

$$\frac{\partial^2}{\partial h^2} Q_\nu(h) = \nu^{-1} I_\nu(\theta_\nu(h),\tau),$$

$$\frac{\partial^3}{\partial h^3} Q_\nu(h) = \nu^{-\frac{3}{2}} R_\nu(\theta_\nu(h),\tau),$$

where U_ν, I_ν and R_ν are given by (3.8)–(3.10). Hence we get the Taylor expansion:

$$\begin{aligned} Q_\nu(h) &= \nu^{-\frac{1}{2}} h^T U_\nu(\theta_0,\tau) \\ &+ \frac{1}{2}\nu^{-1} h^T I_\nu(\theta_0,\tau)\, h \\ &+ \frac{1}{6}\nu^{-\frac{3}{2}} h^T R_\nu(\theta_\nu^*,\tau)\, h^{<2>}, \end{aligned}$$

where θ_ν^* is somewhere on the line-segment between θ_0 and $\theta_\nu(h)$. In the proofs of consistency and asymptotic normality (see Van Pul (1990)) it is deduced that:

$$\nu^{-\frac{1}{2}} U_\nu(\theta_0,\tau) \rightarrow_D \mathcal{N}(0,\Sigma),$$

$$\nu^{-1} I_\nu(\theta_0) \rightarrow_p -\Sigma,$$

$$\nu^{-\frac{3}{2}} R_\nu(\theta_\nu^*,\tau) \rightarrow_p 0,$$

as $\nu\rightarrow\infty$, for all sequences (θ_ν^*) converging to θ_0. Hence, this yields us exactly the local asymptotic normality (LAN) property (3.13). This proves part (iii) of Theorem 3.1. □

(iv) Asymptotic efficiency of the ML-estimators. In the proof of asymptotic normality of $\hat{\theta}_\nu$ (Van Pul (1990)), it is derived that:

$$\nu^{-\frac{1}{2}} U_\nu \rightarrow_{D(\theta_\nu)} \mathcal{N}(0,\Sigma), \tag{3.32}$$

$$\sqrt{\nu}(\hat{\theta}_\nu-\theta_0) - \Sigma^{-1}\nu^{-\frac{1}{2}} U_\nu \rightarrow_{D(\theta_0)} \mathcal{N}(0,\Sigma^{-1}). \tag{3.33}$$

Moreover, the LAN-property of the model, proved in (iii), gives us by using (3.32):

$$\log \frac{d\mathbb{P}_{\theta_\nu}}{d\mathbb{P}_{\theta_0}} \to_{D(\theta_0)} \mathcal{N}(-\frac{1}{2}h^T\Sigma h,\ h^T\Sigma h).$$

Hence:

$$\begin{bmatrix} \sqrt{\nu}(\hat{\theta}_\nu - \theta_0) \\ \log \frac{d\mathbb{P}_{\theta_\nu}}{d\mathbb{P}_{\theta_0}} \end{bmatrix} \to_{D(\theta_0)} \mathcal{N}\left(\begin{bmatrix} 0 \\ -\frac{1}{2}h^T\Sigma h \end{bmatrix}, \begin{bmatrix} \Sigma^{-1} & h \\ h^T & h^T\Sigma h \end{bmatrix} \right).$$

From Le Cam's third lemma (see van der Vaart (1987), pp. 180-181), we can now conclude the contiguity of $\mathbb{P}_{\theta_\nu}$ and $\mathbb{P}_{\theta_0}$ and we have:

$$\sqrt{\nu}(\hat{\theta}_\nu - \theta_0) \to_{D(\theta_\nu)} \mathcal{N}(h,\ \Sigma^{-1})$$

and thus

$$\sqrt{\nu}(\hat{\theta}_\nu - \theta_\nu) = \sqrt{\nu}(\hat{\theta}_\nu - (\theta_0 + \nu^{-\frac{1}{2}} h)) \to_{D(\theta_\nu)} \mathcal{N}(0,\ \Sigma^{-1}). \qquad (3.34)$$

Combining (3.33) and (3.34), we see that for all $h \in \Theta$:

$$\lim_{\nu\to\infty} \mathcal{L}_{\theta_\nu}\left[\sqrt{\nu}(\hat{\theta}_\nu - \theta_\nu) \right] = \lim_{\nu\to\infty} \mathcal{L}_{\theta_0}\left[\sqrt{\nu}(\hat{\theta}_\nu - \theta_0) \right];$$

this means by definition the regularity of the maximum likelihood estimator $\hat{\theta}_\nu$. Now we use an appropriate version of the well-known convolution theorem (see van der Vaart (1987)), which states in our case that the limit-distribution of any regular estimator $\tilde{\theta}_\nu$ of θ_0 satisfies

$$\lim_{\nu\to\infty} \mathcal{L}_{\theta_0}\left[\sqrt{\nu}(\tilde{\theta}_\nu - \theta_0) \right] = \mathcal{N}(0,\ \Sigma^{-1}) * \mathcal{M}_{\theta_0}. \qquad (3.35)$$

Because (3.33) implies that for $\tilde{\theta}_\nu := \hat{\theta}_\nu$ we get $\mathcal{M}_{\theta_0} \equiv 0$ in (3.35), we have proved that the maximum likelihood estimator $\hat{\theta}_\nu$ is asymptotically efficient. This proves part (iv) and hence completes the proof of Theorem 3.1. □

3.4 An application to software reliability theory

Several statistical models have been proposed in order to estimate the evolution in reliability of computer software during the debugging phase. In the introduction we introduced the Jelinski-Moranda model. In this section we present another well-known model in the theory of software reliability, namely the Littlewood model (1980), and we discuss a generalisation of this model. Other well-known software reliability models that fit in our framework are the model of Goel & Okumoto (1979) and the Poisson-Gamma model discussed in Koch & Sprey (1983). For backgrounds and notation we refer to Example 3.1 in the introduction of this chapter.

Example 3.2 *The Littlewood model.* Recall that the failure intensity in the Jelinski-Moranda model is given by

$$\lambda^{\mathrm{JM}}(t) = \phi_0 \left[N_0 - n(t-) \right], \; t \in [0,\tau]. \tag{3.36}$$

Also in the model, introduced by Littlewood (1980), it is assumed that at any time the failure rate is proportional to the number of remaining errors. The main difference in the Littlewood model with respect to the Jelinski-Moranda model, is the fact that each fault does not make the same contribution to the failure rate $\lambda(t)$. Littlewood's argument for that is that larger faults will produce failures earlier than smaller ones. He treats ϕ_j, the failure rate of fault j, as a stochastic variable and suggests a Gamma distribution:

$$\phi_j \sim \Gamma(a_0, b_0), \quad j = 1, \ldots, N.$$

Defining the expected occurrence rate of faults not occurred up to time t, as

$$\phi(t) := \mathbb{E}\, \{ \phi_j \,|\, T_j > t \},$$

with

$$\phi_j \sim \Gamma(a_0, b_0),$$

$$T_j \,|\, \phi_j = \phi \sim \exp(\phi),$$

a simple calculation yields

$$\phi_j \,|\, T_j > t \sim \Gamma(a_0, b_0 + t)$$

and hence

$$\phi(t) = \frac{a_0}{b_0 + t}.$$

An application of the so called *innovation theorem* (Aalen (1978), Bremaud (1977)) now shows that the failure intensity of the software at time t is given by

$$\lambda^{L}(t)=\frac{a_0\left[N_0-n(t-)\right]}{b_0+t}. \tag{3.37}$$

By a simple reparametrisation, namely

$$\rho_0=\frac{1}{a_0},\quad \mu_0=\frac{b_0}{a_0},$$

we get from (3.37):

$$\lambda^{GL}(t)=\frac{N_0-n(t-)}{\mu_0+\rho_0 t}, t\in[0,\tau], \tag{3.38}$$

where *GL* stands for *generalised Littlewood.* Actually this provides an extension of the Littlewood model, allowing also small values of $\rho_0 \leq 0$. Note that when $\rho_0=0$ we are dealing with the model we discussed earlier, namely the Jelinski-Moranda model. We can therefore treat the Jelinski-Moranda model as another special (limit) case of the Littlewood model. Note that both the Littlewood and the Jelinski-Moranda model have as a special case the Poisson model (with constant failure intensity); this is the limit case letting $N_0 \rightarrow \infty$ and $\mu_0 \rightarrow \infty$ in (3.38) such that N_0/μ_0 is a constant. □

Remark 3.2 The models studied by Aalen (1980) are of the form

$$\lambda_j(t) = \sum_{i=1}^{p} \alpha_i Y_{ij}(t),\ j=1...n,$$

where the $\alpha_i(t)$ are non-parametric functions of t and the $Y_{ij}(t)$ are arbitrary observable processes. It should be noticed that (3.38) can be written as:

$$\lambda^{GL}(t) = \frac{N_0}{\mu_0+\rho_0 t}1 + \frac{-1}{\mu_0+\rho_0 t}n(t-), t\in[0,\tau],$$

where 1 and $n(t-)$ are indeed observable processes. In contrast to the models of Aalen, in software reliability models the coefficients $\alpha_i(t)$ are parametric functions of time (and sometimes even constant in time). Additionally, in software reliability models typically $n=1$, while Aalen's n is large. This calls for models and methods different from the typical Aalen type ones. □

Let us now consider how we apply asymptotic theory to the generalised model given by (3.38). Letting $N = \nu\gamma$ conceptually increase as described in Section 3.2, we see that the corresponding sequence of intensity functions can be written in the standard form (3.5):

$$\lambda_\nu^{GL}(t,\theta) = \nu\beta^{GL}(t,\theta,x_\nu),$$

where $\theta=(\gamma,\mu,\rho)$, x_ν is given by (3.4) and

$$\beta^{GL}(t,\ (\gamma,\mu,\rho),\ x) = \frac{\gamma - x(t-)}{\mu + \rho t}, \tag{3.39}$$

is defined on $[0,\tau]\times(\Theta\times K)$, where

$$\Theta\times K := \{\ (\gamma,\mu,\rho,x)\in\mathrm{IR}^+\times\mathrm{IR}^+\times\mathrm{IR}\times D([0,\tau]) : \mu+\rho\tau>0,\ 0\le x(t)\le\gamma,\ t\in[0,\tau]\ \}.$$

As an application we show that the results of Theorem 3.1 hold both for the Jelinski-Moranda and the Littlewood model.

Theorem 3.2 *Let* $\tau>0$. *We assume that the failure data are generated by the intensity function* β *in (3.39) with true parameter value* $\theta_0:=(\gamma_0,\mu_0,\rho_0)$ *satisfying* $\gamma_0>0$ *and* $\mu_0+\rho_0 t>0, t\in[0,\tau]$. *If we define for* $t\in[0,\tau]$

$$x_0(t) := \begin{cases} \gamma_0\left[1-e^{-t/\mu_0}\right] & ,\ \text{if } \rho_0 = 0, \\ \gamma_0\left[1-\left(\dfrac{\mu_0}{\mu_0+\rho_0 t}\right)^{1/\rho_0}\right] & ,\ \text{if } \rho_0 \neq 0, \end{cases}$$

then β *satisfies conditions (G1)–(G2) & (L1)–(L3) and hence*

(i) *With probability tending to one, the likelihood equations have exactly one consistent solution* $\hat\theta$.
(ii) *A consistent solution* $\hat\theta$ *is asymptotically normal and efficient.*
(iii) *The model satisfies the LAN-property.*

We here present a verification of (L3), which is more elegant, than the one given in Van Pul (1990). For full details, however, of the verification of (G1)–(G2) and (L1)–(L2), which is technical but routine, we refer to Van Pul (1990). It is shown there that the following choice of Θ_0 and K_0 will be appropriate:

$$\Theta_0 := [\varepsilon_\gamma, M_\gamma] \times [\varepsilon_\mu, M_\mu] \times [\varepsilon_\rho, M_\rho],$$

$$K_0 := \{\ x\in K : \|x-x_0\|_{\sup} \le \varepsilon_x\ \},$$

where

$$0<\frac{1}{2}[\gamma_0+x_0(\tau)]<\varepsilon_\gamma<\gamma_0<M_\gamma,$$

$$0<\varepsilon_\mu<\mu_0<M_\mu, \quad -\frac{\mu_0}{\tau}<\varepsilon_\rho<\rho_0<M_\rho \quad \text{and} \quad 0<\varepsilon_x<\frac{1}{2}[\gamma_0-x_0(\tau)],$$

with $\varepsilon_\rho < 0 < M_\rho$.

Proof (L3) To verify (L3) one may check for instance that det $\Sigma \neq 0$; this is, however, extremely tedious. See Van Pul (1990). A much simpler approach is to note that Σ is the covariance matrix of

$$d_1 := \int_0^\tau \frac{1}{N-n(t-)} \mathrm{d}M(t),$$

$$d_2 := \int_0^\tau \frac{1}{\mu+\rho t} \mathrm{d}M(t)$$

and

$$d_3 := \int_0^\tau \frac{t}{\mu+\rho t} \mathrm{d}M(t).$$

Therefore, we have det $\Sigma=0$ if and only if there exist coefficients a, b and c (not all equal to zero), such that

$$D := ad_1 + bd_2 + cd_3 \tag{3.40}$$

is constant. Now we consider the following two cases. Firstly, with positive probability $n(t)$ makes no jump at all in $[0,\tau]$. Secondly, also with positive probability, the process $n(t)$ makes exactly one jump in $[0,\tau]$. Suppose this jump is at time T. One can easily check that in the case of no jump (3.23) is given by

$$D_0 = N\int_0^\tau \frac{b+ct}{(\mu+\rho t)^2}\,\mathrm{d}t - \frac{a}{\rho}\log(1+\frac{\rho}{\mu}\tau)$$

and in the case of exactly one jump at T by

$$D_1(T) = \left[\frac{a}{N} + N\int_0^\tau \frac{b+ct}{(\mu+\rho t)^2}\,\mathrm{d}t - \frac{a}{\rho}\log(1+\frac{\rho}{\mu}\tau)\right] + \frac{b+cT}{\mu+\rho T} - \int_T^\tau \frac{b+ct}{(\mu+\rho t)^2}\,\mathrm{d}t.$$

Obviously, D_1 is a non-constant function of T, when b and c are not both equal to zero. But given $b=0$, $c=0$ we see that the constants D_0 and D_1 are different, except for the degenerate case that also $a=0$. We have hence proved that there do not exist coefficients a, b and c (not all three equal to zero), such that D is constant. This yields the non-singularity of Σ. □

Remark 3.3 The software reliability models, where $\theta=(\gamma,\psi)$ and $N=\nu\gamma$ is a parameter of interest, typically satisfy the rescaling (homogeneity) condition

$$\beta(t;(c\gamma,\psi);cx) = c\ \beta(t;(\gamma,\psi);x), \tag{3.41}$$

for all $c>0$. It is easy to check that the asymptotic variance of $\hat{\psi}$ now does not depend on γ, while that of $\hat{\gamma}$ is proportional to γ. This guarantees that asymptotic confidence intervals for N and ψ do not depend on the (arbitrary) choice of ν. □

3.5 Some numerical results

We now present a study of the behaviour of the ML-estimators in practice, computed from both real data and and from simulated data generated by the Jelinski-Moranda model. The simulation results of Van Pul (1991a) confirm the asymptotic theory as derived in this chapter. They also show on the other hand that the convergence in distribution is rather slow and that for small values of N_0 the distributions of $\hat{N}$ and $\hat{\phi}$ can be very skew. With use of the Wilks likelihood ratio test statistic (3.14), however, we were able to build confidence intervals for the model parameters that are much more satisfactory than intervals based on the approximate normal test statistic.

In this section we discuss two numerical examples. In Example 3.3, which deals with some real data collected by Moek (1983), the Jelinski-Moranda model and the Littlewood model are compared. In Example 3.4, data are simulated according to the Jelinski-Moranda model. The theoretical asymptotic normality is studied and coverage percentages of confidence intervals based on the asymptotic normal and on the WLRT statistic are compared. More background, calculations and detailed results, both on Moek's data and on the simulated data, can be found in Van Pul (1991a).

Example 3.3 *A case study.* The models of Jelinski-Moranda and Littlewood have been applied to real data from Project A, concerning an information system for registering aircraft movements. For more details see Moek (1983, 1984). Failure data collected during the testing stage (in the operational environment) are given in Table 3.1. Note that there is a misprint in T_{37} of the original data in Moek (1983). Furthermore, T_{44} does not represent a failure time, but is assumed to be the stopping time τ of the testing process. Figure 3.1 gives the counting process associated with the data of Table 3.1. We calculated maximum likelihood estimators for the model parameters of the models of Jelinski-Moranda (JM), Littlewood (L) and Generalised Littlewood (GL). Their intensity functions are given by respectively (3.36), (3.37) and (3.38). To determine MLE's for the (three-parameter) models L and GL we used a standard optimisation program, written in Pascal. This program, called *Amoeba* and described in Vetterling *et al.* (1985), carries out a down-hill simplex method. The results are given in Table 3.2.

i	t_i	T_i	i	t_i	T_i
1	0.00088	0.00088	23	0.00445	0.13321
2	0.00343	0.00431	24	0.00486	0.13807
3	0.00286	0.00717	25	0.00064	0.13871
4	0.01176	0.01893	26	0.00399	0.14270
5	0.00475	0.02368	27	0.02684	0.16954
6	0.00024	0.02392	28	0.00227	0.17181
7	0.00230	0.02622	29	0.00020	0.17201
8	0.00857	0.03479	30	0.03918	0.21119
9	0.00462	0.03941	31	0.01491	0.22610
10	0.00106	0.04047	32	0.01467	0.24077
11	0.00382	0.04429	33	0.01631	0.25708
12	0.01480	0.05909	34	0.03841	0.29549
13	0.00177	0.06086	35	0.00112	0.29661
14	0.02427	0.08513	36	0.03056	0.32717
15	0.00480	0.08993	37	0.00621	0.33338
16	0.00047	0.09040	38	0.00012	0.33350
17	0.00004	0.09044	39	0.02021	0.35371
18	0.01017	0.10061	40	0.02640	0.38011
19	0.00112	0.10173	41	0.03780	0.41791
20	0.00098	0.10271	42	0.07422	0.49213
21	0.02430	0.12701	43	0.08444	0.57657
22	0.00175	0.12876	44	0.02343	0.60000

Table 3.1
Failure times for Moek's project A (CPU time in Msec).

	JM	L	GL
max log L_τ	156.2290	156.2298	156.8618
$\hat{N}$	44.0734	44.0742	43.0000
$\hat{\phi}$	5.5465	5.5463	-
$\hat{a}$	-	∞	-3.9246
$\hat{b}$	-	∞	-0.8191
$\hat{\mu}$	-	0.1803	0.2087
$\hat{\rho}$	-	0.0000	-0.2548
$\hat{\lambda}(\tau)$	5.9536	5.9578	0.0000

Table 3.2
Comparison of maximum likelihood estimators for the models of Jelinski-Moranda (JM), Littlewood (L) and Generalised Littlewood (GL) with use of data from Table 3.1.

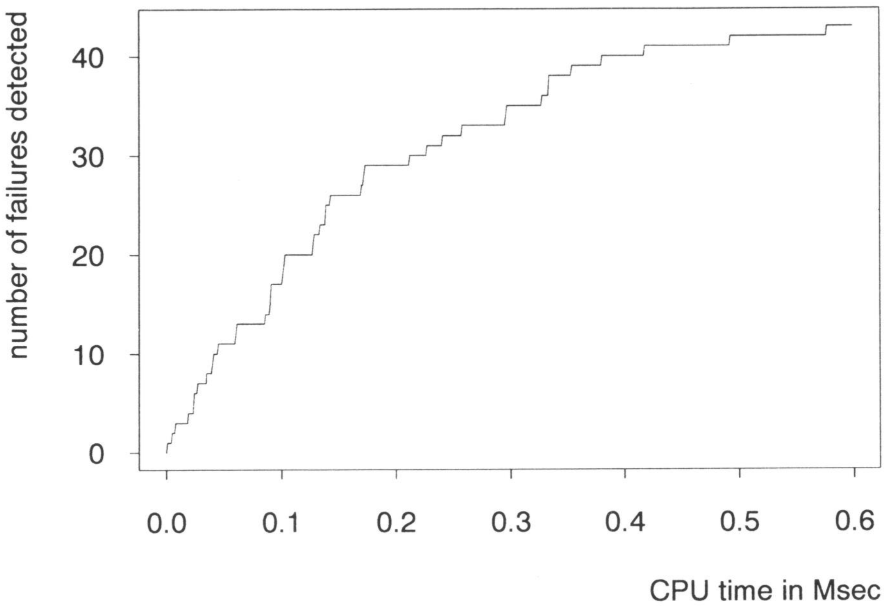


Figure 3.1
Counting process belonging to failure data of project A.

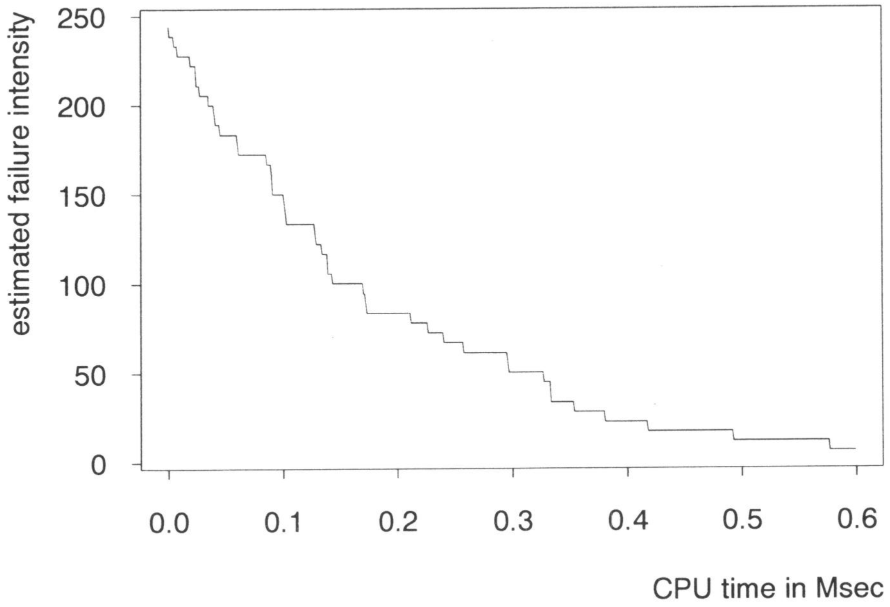


Figure 3.2
Estimated failure intensity of Jelinski-Moranda and Littlewood model.

We find that conditioned on $\rho \geq 0$, the log-likelihood function of the Littlewood model is maximal for $\rho=0$. In this case the Littlewood model reduces to the Jelinski-Moranda model with $\phi=1/\mu=5.5463$. In the generalised Littlewood model (3.38), also allowing small negative values for ρ, the log-likelihood function is maximised for $N=43.0000$ and hence $\lambda=0$. This seems not to make much sense. It should be noticed that the number of bugs is too small to make accurate predictions. This will be pointed out in the next example. The estimated failure intensity of the Jelinski-Moranda and Littlewood model ($\rho \geq 0$) are identical and given in Figure 3.2. More results on this case study, including standard deviations etc. can be found in Andersen *et al.* (1993). □

Example 3.4 *Simulation of the Jelinski-Moranda model.* In our simulation experiments we generated failure times according to the Jelinski-Moranda model with $\phi_0=1$, $\tau=1$ and various values for N_0. From the asymptotic theory developed in Section 3.3 and 3.4, it follows that we can define centered and normalised quantities $\hat{X}$ and $\hat{Y}$, satisfying:

$$\hat{X} := \frac{\hat{N}-N_0}{\sqrt{N_0}} \rightarrow_D \mathcal{N}\left(0\,,\ \frac{1-e^{-\phi_0\tau}}{e^{\phi_0\tau}+e^{-\phi_0\tau}-\tau^2\phi_0^2-2}\right), \tag{3.42}$$

$$\hat{Y} := \sqrt{N_0}\,(\hat{\phi}-\phi_0) \rightarrow_D \mathcal{N}\left(0\,,\ \frac{\phi_0^2(e^{\phi_0\tau}-1)}{e^{\phi_0\tau}+e^{-\phi_0\tau}-\tau^2\phi_0^2-2}\right), \tag{3.43}$$

as $N_0 \rightarrow \infty$. We generated and studied sets of 10,000 replicates of $\hat{X}$ and $\hat{Y}$. The figures for $N_0=50$, 500 and 5,000 are given in Table 3.3.

N_0	Mean $\hat{X}$	Var $\hat{X}$	Skew $\hat{X}$	Mean $\hat{Y}$	Var $\hat{Y}$	Skew $\hat{Y}$
50	-0.2270	4.4943	1.6352	2.1887	15.1202	0.6342
500	0.2884	10.0716	0.9480	0.5038	20.1352	0.0351
5,000	0.1145	7.4131	0.3837	0.1071	19.7273	0.0178
∞	0.0000	7.3365	0.0000	0.0000	19.9423	0.0000

Table 3.3
Means, variances and skewness coefficients of
simulated approximate normal quantities X and Y.
Number of replicates $K=10{,}000$, $\phi=1$, $\tau=1$.

We see that the convergence of $\hat{X}$ and $\hat{Y}$ to normal distributions with mean zero and asymptotic variances as expected (in (3.42) and (3.43)) is rather slow. The difference in the asymptotic behaviour of $\hat{N}$ and $\hat{\phi}$ is illustrated by the histograms and qq plots given in Figures 3.3 and 3.4. Both tables and figures give the same impression, namely that the distribution of $\hat{N}$ shows a severe skewness and that the distribution of $\hat{\phi}$ is rather biased for small N_0. Both defects slowly disappear as N_0 increases.

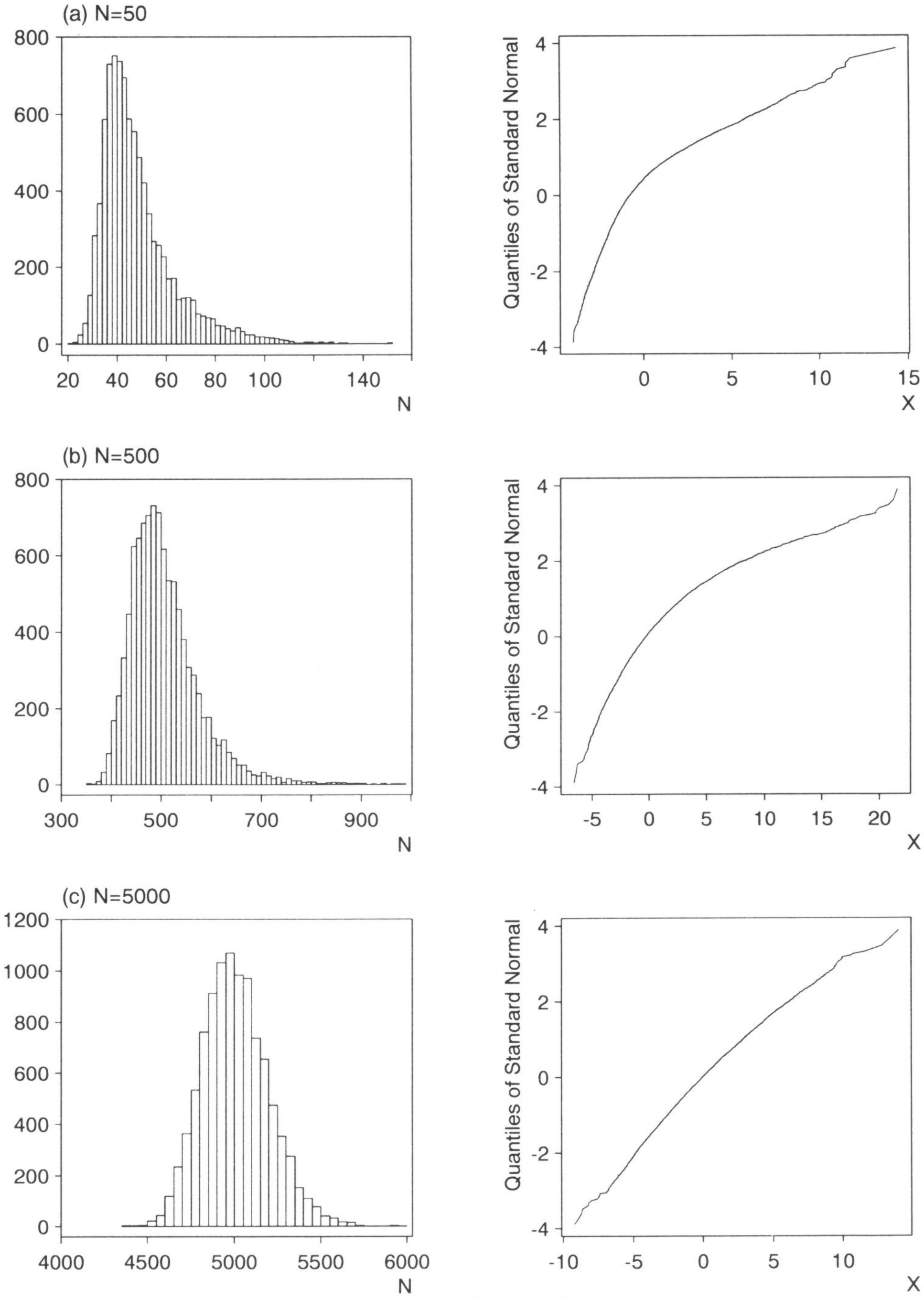


Figure 3.3
Histogram of $\hat{N}$ and qq plot of $\hat{X}$:
(a) N_0=50, (b) N_0=500, (c) N_0=5,000.

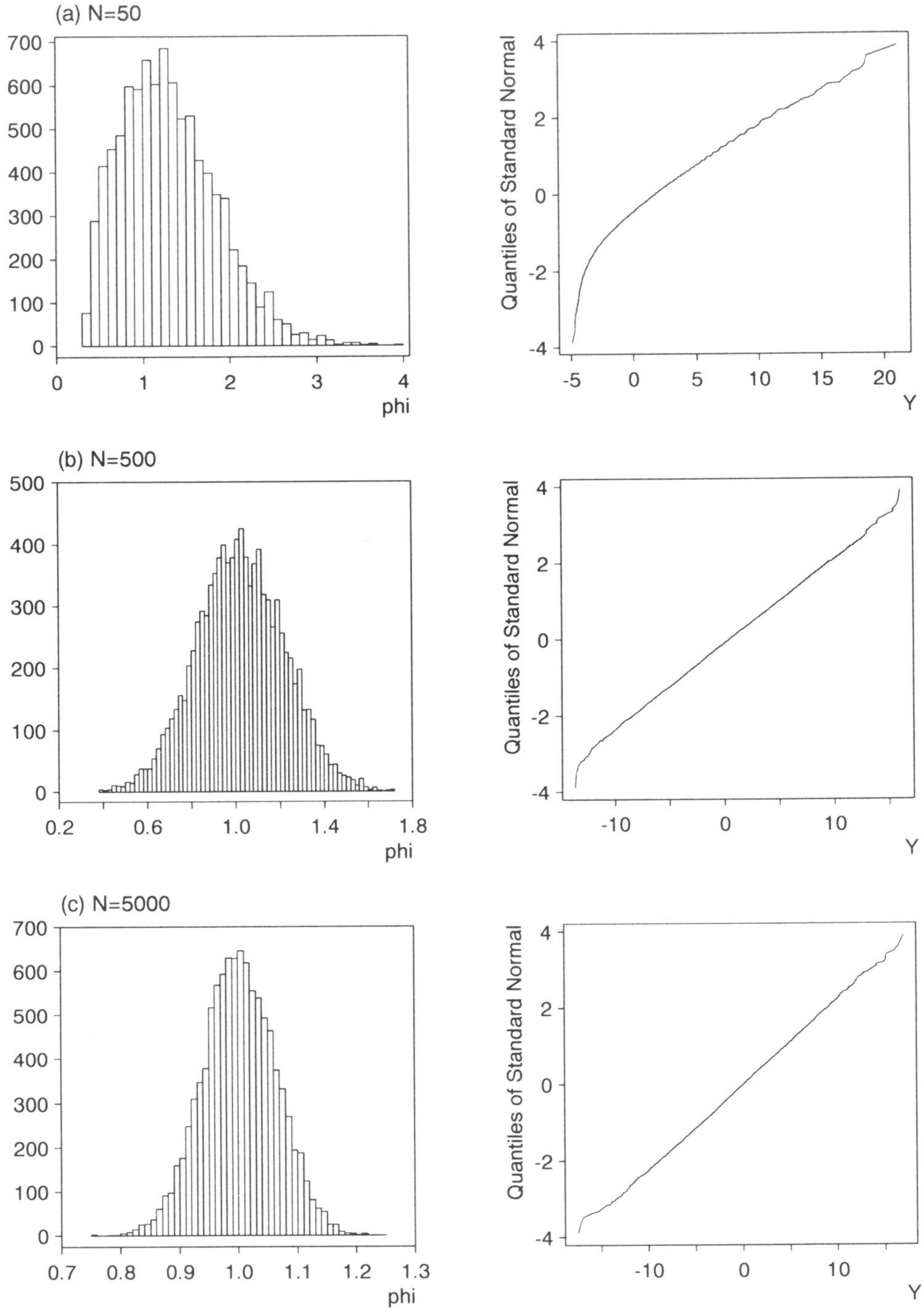


Figure 3.4
Histogram of $\hat{\phi}$ and qq plot of $\hat{Y}$:
(a) N_0=50, (b) N_0=500, (c) N_0=5,000.

As the distribution of $\hat{N}$ is skew, the coverage percentages of confidence intervals based on the asymptotic normal statistic could be expected to be disappointing. In Table 3.4 we compare these percentages with those based on the Wilks test statistic (3.14).

	Approximate normal			Wilks		
α	N_0=50	N_0=500	N_0=5,000	N_0=50	N_0=500	N_0=5,000
50	55.58	50.90	50.43	52.56	50.43	50.39
60	60.03	61.28	60.62	62.38	59.95	60.12
70	64.15	71.81	70.95	71.76	70.27	70.75
80	68.86	81.82	80.71	80.45	80.26	80.48
90	74.52	88.42	90.30	89.64	89.79	90.32
95	77.62	91.30	94.99	94.16	94.82	95.23

Table 3.4
Coverage probabilities of the two-sided approximate normal
and the two-sided Wilks confidence intervals based on simulated data of Table 3.3.
Number of replicates K=10,000, ϕ=1, τ=1.

As the Wilks confidence intervals are larger, shifted to the right (and hence not symmetric around N_0) in comparison with the approximate normal confidence intervals, for high levels of confidence the Wilks intervals are significantly better (and have coverage probabilities that are less skew) than the approximate normal ones (see Van Pul (1991a)). □

3.6 Concluding remarks, future investigations and open problems

As stated in Remark 3.2, Theorems 3.1 and 3.2 remain valid if we replace (G1)—(G2) & (L1)—(L3) by the weaker set of conditions (C1)—(C4). Conditions comparable to these ones are also given by Cramér (1946) and Kulldorff (1957), using classical statistical techniques to prove consistency and asymptotic normality of maximum likelihood estimators. Nowadays modern methods have been developed by Ibragimov & Khas'minskii (1979), Jacod & Shiryaev (1988), Dzhaparidze & Valkeila (1988) and Le Cam & Yang (1990) among others leading to the same results (and even more), without requiring the existence of higher derivatives of the intensity function and so weakening condition (L1) (and (C2)) considerably. Ibragimov & Khas'minskii (1979) consider the parametric case, but no theory for counting processes is developed, while Jacod & Shiryaev (1988) and Dzhaparidze & Valkeila (1988) study only binary experiments for counting processes. Also the work of Gill (1980) and van der Vaart (1987) should be mentioned here. Therefore it seems very plausible that such methods can be applied also in our case. Indeed, the assumption of existence of the third derivative of β with respect to θ can be abandoned (and for consistency even the existence of the second derivative!). Other conditions on β—maybe weaker, but harder to verify—will replace them. In commonly used models, however, intensity functions tend to be very smooth and determining the existence of derivatives (with respect to θ) is relatively easy.

Moreover, we think we can improve the construction of confidence intervals by making use of parametric bootstrap methods. The validity of the parametric bootstrap method will follow by standard arguments on contiguity, regular estimators and the Skorohod-Dudley-Wichura almost sure representation theorem (see Gill (1989)). Asymptotic consistency of the parametric bootstrap is proved in Van Pul (1991c) and some numerical results in software reliability are presented. [Chapter 4 of this thesis.]

Furthermore, we note that Theorem 3.1 does not claim that the maximum likelihood equations have a unique solution. It only states that with a probability tending to one, among all these solutions, only one of them will be consistent. Moek (1983) developed an easy criterion, satisfied with probability tending to one, for the existence of a unique solution of the ML equations for the Jelinski-Moranda model. The problem in case of the Littlewood model, however, is much harder and is in fact still an open question. Instead of finding such a criterion in the Littlewood case, Barendregt and Van Pul (1991) developed an algorithm in order to determine the consistent one from a set of solutions from the ML equations by choosing the nearest solution to a consistent estimator. A more general approach may be possible, probably with use of compactification ideas (see Bahadur (1967)). [Chapter 5 of this thesis.]

Another topic of future investigation will be the study of goodness of fit tests. We intend to follow the martingale approach of Khmaladze (1981). See also Geurts *et al.* (1988) and Hjort (1990). [Chapter 6 of this thesis.]

Finally, of course, our ultimate goal will be to study more realistic models, incorporating imperfect repair and software growth and taking account of covariate measurements. We have recently constructed such a model (Van Pul (1991b)) and are now investigating whether this model fits in the theory developed so far. [Chapter 7 of this thesis.]

Chapter 4

Simulations and the Bootstrap*

4.1 Introduction

Several investigators have built statistical models in order to estimate the evolution in reliability of computer software during the debugging phase. We refer to Musa *et al.* (1987) for a complete overview of the most common software reliability models. A lot of those software reliability models can be formulated in terms of counting processes, counting the number of failure occurrences. We are interested in the asymptotic properties of the maximum likelihood estimation method for parametric counting process models. A novel aspect of our approach is the fact that—in order to treat asymptotic theory—instead of increasing the time variable or the number of data as is usually the case, we will (conceptually) increase one of the model parameters itself. To illustrate the problem and to motivate our concepts, we will present here one of the oldest and most elementary software reliability models, namely that of Jelinski-Moranda (1972), as an example.

Example 4.1 *The Jelinski-Moranda model.* A computer program has been executed during a specified exposure period and the interfailure times are observed. The repairing of a fault takes place immediately after it produces a failure and no new faults are introduced with probability one. Let N the unknown number of faults initially present in the software. Let the exposure period be $[0,\tau]$ and let $n(t)$, $t \in [0,\tau]$, denote the number of faults detected up to time t. Define $T_0 := 0$ and let T_i, $i=1,2,...,n(\tau)$, the failure time of the i-th occurring

* This chapter is based on CWI-reports BS-9122 and BS-9123 and on the paper "Simulations on the Jelinski-Moranda model of software reliability; application of some parametric bootstrap methods" by Mark C. van Pul, which appeared in *Statistics and Computing* (1992), **2,** pp. 121-136.

failure, while $t_i := T_i - T_{i-1}$, $i=1,2,...,n(\tau)$, denotes the interfailure time, that is the time between the i-th and the $(i-1)$-th occurring failure. Finally we define $t_{n(\tau)+1} := \tau - T_{n(\tau)}$. In the Jelinski-Moranda model, introduced in 1972 and a few years later generalised by Musa (1975), the failure rate of the program is at any time proportional to the number of remaining faults and each fault still present makes the same contribution to the failure rate. So if $(i-1)$ faults have already been detected, the failure rate for the i-th occurring failure, λ_i, becomes

$$\lambda_i = \phi_0 \left[N_0 - (i-1) \right], \tag{4.1}$$

where ϕ_0 is the true failure rate per fault (the occurrence rate) and N_0 is the true number of faults initially present in the software. In terms of counting processes we can write

$$\lambda^{\mathrm{JM}}(t) = \phi_0 \left[N_0 - n(t-) \right], \; t \in [0,\tau], \tag{4.2}$$

where $\lambda(t)$, $t \in [0,\tau]$ denotes the failure rate at time t. The interfailure times $t_i, i=1,...,n(\tau)$, are independent and exponentially distributed with parameter λ_i given by (4.1). By using the information obtained from the test experiment one can estimate the the parameters of the underlying model, especially N_0, the total number of faults initially present in the software. Mostly maximum likelihood estimation (MLE) is used for this purpose. Now let us look at the way we will treat asymptotic behaviour. It does not make sense to let τ, the stopping time, grow to infinity. In the long run the estimate of the total number of faults will trivially be equal to the true number of faults. It makes more sense to (conceptually) increase the number of faults in the program. The idea is that then asymptotics should be relevant to the practical situation in which N_0 is large and $n(\tau)/N_0$ not close to zero or one. □

Remark 4.1 The Jelinski-Moranda model is perhaps not the most realistic model for computer systems; however, it represents a natural starting point for such a study. The model is very popular, although practical application has often been disappointing. More sophisticated models often do not give a better fit to the data. Beside, the conclusion that the JM-model is not satisfied can itself provide valuable information; especially if the cause of this can be identified. □

In an earlier paper (Van Pul (1990)) we derived theoretical results for the asymptotic behaviour of the MLE-procedure for a large class of software reliability models. For a general class of intensity functions, it was proved that the maximum likelihood estimators for the model parameters are asymptotically normally distributed. This enables us to use the classical approximate normal confidence intervals. In Van Pul (1990) we also suggested using intervals based on the asymptotically equivalent Wilks Likelihood Ratio Test (WLRT) statistic, which is χ^2-distributed (see Rao (1973)).

In this chapter we study how well these theoretical results appear in practice for the Jelinski-Moranda model by a simulations study using data generated according to this model. We compute point estimates, (upper) confidence bounds and confidence intervals for the model parameters and study their asymptotic behaviour. For the application of the Jelinski-Moranda model (and other models) to real data we refer to Moek (1983, 1984) and Andersen *et al.* (1993). It turns out that the theoretical convergence to normality is rather slow and that empirical distributions of the MLE's for small N_0 tend to be very skew. This leads of course to systematic errors in the coverage percentages of confidence intervals based on the classical normal test statistic. Confidence intervals based on the Wilks Likelihood Ratio Test (WLRT) statistic are much better, but still have asymmetrical coverage probabilities. Moreover, upper-bounds based on the WLRT statistic for small N_0 (that is, when the asymptotic normal behaviour has not occurred yet) are often infinite and therefore in some sense less informative. On the other hand, the fact that the data cannot exclude the possibility of constant (or even increasing) failure intensity, could be important to know in practical situations. A third approach to construct confidence intervals for the model parameters is investigated. The idea is to apply a parametric bootstrap method on standardised or studentised versions of the approximate normal test statistic or to estimate the second order term in the Edgeworth expansion of the distribution of the MLE's (Hall (1988)).

In the next section of this chapter we introduce the general framework of counting processes and intensities as described in Van Pul (1990) and discuss how parametric bootstrap methods could be applied. We show that in this general setting the parametric bootstrap 'works', that is, is asymptotically consistent. The proof which makes use of contiguity-arguments and the Skohorod-Dudley-Wichura almost-sure-representation theorem (see for instance Pollard (1989)), is based on stronger versions of the concepts of local asymptotic normality (LAN) and regularity. In Section 4.3 we consider the simulation experiments more closely. We describe the computations of the MLE's in Section 4.4 and discuss several ways to construct confidence intervals in Section 4.5. In the sixth section we discuss the results of computations, estimations and statistical methods (like the parametric bootstrap) applied on simulated data according to the model of Jelinski-Moranda, as described in Example 4.1. We close this chapter with some concluding remarks in Section 4.7.

4.2 The general framework

[First we summarise the main results of Chapter 3.] Let a counting process $n(t)$ be given. Only during a specific time interval $[0,\tau]$, are jumps of the counting process $n(t)$ observed. In this chapter we will assume that the intensity function associated with the counting process exists and is a member of some specified parametric family, that is:

$$\lambda(t) \ := \ \lambda(t;N,\psi),$$

with $t \in [0,\tau]$, $N \in \mathbb{N}$, $\psi \in \Psi$ and $\Psi \subset \mathbb{R}^{p-1}$ for an integer p. Let N_0 and ψ_0 be the true parameter values. Typically the parameter N_0 represents the scale or size of the problem (sometimes $N_0 = n(\infty)$), while ψ_0 is a nuisance vector parameter. We will be interested in estimation of N_0 and ψ_0 as $N_0 \to \infty$. We assume that the model is also meaningful for non-integer N. For instance the intensity function (4.2) of the Jelinski-Moranda model can be generalised to

$$\lambda^{\mathrm{JM}}(t) = \phi\left[N - n(t-)\right] I\left\{ n(t) < N \right\},$$

where $t \in [0,\tau]$ and $I\{.\}$ denotes the indicator function. As we are particularly interested in the parameter estimation when N_0 is large, we will introduce a series of counting processes $n_\nu(t)$, $t \in [0,\tau]$, $\nu = 1,2,\ldots$ and let N conceptually increase. Let $N = N_\nu \rightarrow \infty$ for $\nu \rightarrow \infty$. By the reparametrisation

$$N_\nu = \nu\gamma_\nu$$

with a dummy variable γ_ν, we can denote the associated intensity functions by

$$\lambda_\nu(t;\gamma,\psi) := \lambda(t;\nu\gamma,\psi),$$

with $t \in [0,\tau]$, $\gamma \in \mathrm{IR}^+$, $\psi \in \Psi$, $\nu = 1,2,\ldots.$ Now we consider the estimation of γ and ψ as $\nu \rightarrow \infty$. If the real-life situation has $\nu = N_0$, then $\gamma = \gamma_0 = 1$ and $\psi = \psi_0$. It is rather unorthodox to increase a model parameter itself, in this case N. This complication is solved by estimating γ. We will assume that the maximum likelihood estimators $(\tilde{G}_\nu, \hat{\psi}_\nu)$ for (γ_0, ψ_0) exist. Typically, $(\tilde{G}_\nu, \hat{\psi}_\nu)$ is a root of the likelihood equations

$$\frac{\partial}{\partial(\gamma,\psi)} \log L_\nu(\gamma,\psi;\tau) = 0, \quad \nu = 1,2,\ldots, \tag{4.3}$$

where the likelihood function at time t $L_\nu(\gamma,\psi;t)$ is given by (see Aalen (1978))

$$L_\nu(\gamma,\psi;t) := \exp\left\{\int_0^t \log \lambda_\nu(s;\gamma,\psi)\, \mathrm{d}n_\nu(s) - \int_0^t \lambda_\nu(s;\gamma,\psi)\mathrm{d}s\right\} \tag{4.4}$$

We define for $\nu = 1,2,\ldots$ the stochastic process $x_\nu(t)$ by

$$x_\nu(t) := \nu^{-1} n_\nu(t), \quad t \in [0,\tau].$$

For many models used in practice, this sequence of stochastic processes converges uniformly on $[0,\tau]$ in probability to a deterministic function $x_0(t)$ as $\nu \rightarrow \infty$ (Kurtz (1983)). We assume for this that the counting processes n_ν are generated by associated intensity functions $\lambda_\nu(t)$, satisfying

$$\lambda_\nu(t) \;=\; \nu\beta(t;\theta;x_\nu(t-)),$$

for an arbitrary non-negative and non-anticipating function $\beta:[0,\tau]\times\Theta\times K \rightarrow \mathrm{IR}^+$ satisfying some smoothness conditions, where the model-parameter $\theta = (\gamma,\psi)$ consists of the parameter of most interest γ and a nuisance parameter vector ψ. Under classical smoothness and boundedness conditions on the function β (see for instance Borgan (1984) or Van Pul (1990)), we have the following result:

Theorem 4.1

(i) *Consistency of ML-estimators: With a probability tending to 1, the likelihood equations (4.3) have exactly one consistent solution $\hat{\theta}_\nu$. Moreover this solution provides a local maximum of (4.4).*

(ii) *Asymptotic normality of the ML-estimators: Let $\hat{\theta}_\nu$ be the consistent solution of the maximum likelihood equations (4.3), then*

$$\sqrt{\nu}\,(\hat{\theta}_\nu-\theta_0) \rightarrow_D \mathcal{N}(0,\Sigma^{-1}), \quad \nu\rightarrow\infty,$$

where the matrix $\Sigma=\{\sigma_{ij}(\theta_0)\}$ with for $i,j\in\{1,2,..,p\}$, $\theta\in\Theta_0$: is given by

$$\sigma_{ij}(\theta)=\int_0^\tau \frac{\frac{\partial}{\partial\theta_i}\beta(s,\theta,x_0)\frac{\partial}{\partial\theta_j}\beta(s,\theta,x_0)}{\beta(s,\theta,x_0)}\mathrm{d}s. \tag{4.5}$$

(iii) *Local asymptotic normality of the model: There exist a sequence $U_\nu,\nu=1,2,...$ such that for all $h\in\mathrm{IR}^p$:*

$$\log\frac{dP_{\theta_\nu}}{dP_{\theta_0}} - h^T U_\nu + \frac{1}{2}h^T\Sigma h \rightarrow_{\mathrm{IP}_{\theta_0}} 0$$

for $\nu\rightarrow\infty$, where $\theta_\nu=\theta_0+\nu^{-\frac{1}{2}}h$ and $U_\nu \rightarrow_D \mathcal{N}(0,\Sigma)$, Σ given by (4.5).

(iv) *Asymptotic efficiency of the ML-estimators: $\hat{\theta}_\nu$ is asymptotically efficient in the sense that the limit distribution for any other regular estimator $\tilde{\theta}_\nu$ for θ_0 satisfies*

$$\sqrt{\nu}\,(\tilde{\theta}_\nu-\theta_0) \rightarrow_{D(\theta_0)} Z + Y,$$

where $Z \sim_d \mathcal{N}(0,\Sigma^{-1})$, Z and Y independent. (For a definition of the regularity of an estimator we refer to van der Vaart (1988).)

This theorem and its proof can be found more thoroughly in Van Pul (1990). [End of summary.]

Simulations of the Jelinski-Moranda model will show (see Section 4.4) that asymptotic convergence to the normal distribution is appearing very slowly and that for values of ν not extremely large the empirical distribution functions of the components of $\hat{\theta}_\nu$ can be significantly skew. Hence, confidence intervals based on approximate normal test statistics will turn out to be disappointing. One solution, which is already suggested in Van Pul (1990), is to make use of the Wilks Likelihood Ratio Test Statistic,

$$2\left[\log L_\nu(\hat{\theta}_\nu,\tau) - \log L_\nu(\theta_0,\tau)\right], \tag{4.6}$$

which can be proved to be asymptotically $\chi^2(p)$ distributed and is often thought to have faster convergence.

Another way to deal with deviations from normality is to make use of bootstrap methods. Suppose we want to construct confidence intervals for a one-dimensional real parameter θ. The concept of parametric bootstrapping in the context of software reliability consists of simulating a so called *bootstrap counting process* according to the failure intensity $\lambda(t,\hat{\theta})$, where $\hat{\theta}$ is the maximum likelihood estimator for θ. Repeating this simulation experiment, say M times, we get bootstrap estimators $\hat{\theta}_i^*, i=1,...,M$. We define:

$$G_\nu := \mathcal{L}_{\theta_0}\left[\sqrt{\nu}(\hat{\theta}_\nu-\theta_0)\right], \tag{4.7}$$

$$G_\nu^* := \mathcal{L}_{\hat{\theta}_\nu}\left[\sqrt{\nu}(\hat{\theta}_\nu^*-\hat{\theta}_\nu)\right]. \tag{4.8}$$

We will say that the parametric bootstrap *works* (or is *asymptotically consistent*) if and only if

$$\sup_{x\in\mathbb{R}} \mid G_\nu^*(x) - G_\nu(x) \mid \rightarrow_{P_{\theta_0}} 0. \tag{4.9}$$

See also Bickel & Freedman (1981) and Singh (1981). This result will be derived in the next theorem. Note that, as G_ν converges to a continuous distribution function, a consequence of (4.9) is that confidence intervals for θ based on G_ν^* will have asymptotically the right coverage probabilities, i.e.:

$$\mathbb{P}\left[\hat{\theta}_\nu - \frac{Z_\nu^*(1-\frac{\alpha}{2})}{\sqrt{\nu}} \le \theta_0 \le \hat{\theta}_\nu - \frac{Z_\nu^*(\frac{\alpha}{2})}{\sqrt{\nu}}\right] \rightarrow 1-\alpha,$$

for $\nu \rightarrow \infty$, where $Z_\nu^*(\alpha) := G_\nu^{*\,-1}(\alpha)$. In practice one often uses studentised versions of (4.7) and (4.8),

$$G_\nu^{ST} := \mathcal{L}_{\theta_0}\left[\frac{\sqrt{\nu}(\hat{\theta}_\nu-\theta_0)}{\hat{\sigma}_\nu}\right],$$

$$G_\nu^{ST*} := \mathcal{L}_{\hat{\theta}_\nu}\left[\frac{\sqrt{\nu}(\hat{\theta}_\nu^*-\hat{\theta}_\nu)}{\hat{\sigma}_\nu^*}\right],$$

expecting the second order terms of the Edgeworth expansions to be the same too (see for instance Helmers (1991)). In this chapter we will determine $\hat{\sigma}_\nu$ and $\hat{\sigma}_\nu^*$ simply by substituting respectively $\hat{\theta}_\nu$ and $\hat{\theta}^*_\nu$ for θ_0 in the expected information matrix Σ, given by (4.5). An alternative way to estimate σ consistently, is to make use of the observed information matrix

$$I_\nu(\theta,\tau)\log L_\nu(\theta,\tau). \tag{4.10}$$

Supposing we are in the counting process context, sketched in the beginning of this section (and hence in particular in the software reliability situation of Example 4.1), we can prove (4.9). The following two lemma's will be very useful:

Lemma 4.1 *Under the conditions of Theorem 4.1, we have SLAN (strong local asymptotic normality), that is: there exist a sequence $U_\nu, \nu=1,2,\ldots$ such that for all $h \in \mathbb{R}^p$:*

$$\log \frac{dP_{\theta_\nu}}{dP_{\theta_0}} - h^T U_\nu + \frac{1}{2} h^T \Sigma h \rightarrow_{\mathbb{P}_{\theta_0}} 0,$$

for $\nu \rightarrow \infty$, where $U_\nu \rightarrow_d \mathcal{N}(0,\Sigma)$, Σ given by (4.5), but now with

$$\theta_\nu = \theta_0 + \nu^{-1/2} h + o(\nu^{-1/2}). \tag{4.11}$$

Lemma 4.2 *Under the conditions of Theorem 4.1, asymptotic normality and SLAN imply S-regularity (strong regularity), that is:*

$$\sqrt{\nu}\left(\hat{\theta}_\nu - \theta_\nu\right) \rightarrow_{D(\theta_\nu)} \mathcal{N}(0,\Sigma^{-1})$$

for all sequences θ_ν of the form (4.11).

The proofs of Lemma's 4.1 and 4.2 are slight modifications of the proofs of Theorem 3.1, (iii)&(iv), given in Van Pul (1990), and are therefore omitted here. We are now able to formulate the following result:

Theorem 4.2 *The parametric bootstrap is asymptotically consistent.*

Proof Theorem 4.2 The asymptotic normality of the MLE yields:

$$\lim_{\nu \rightarrow \infty} \sup_{x \in \mathbb{R}} |G_\nu(x) - G(x)| = 0,$$

where $G(x) := \mathcal{N}(0,\Sigma^{-1})$. So to prove (4.9) it is sufficient to show that for all $\varepsilon > 0$:

$$\lim_{\nu \rightarrow \infty} \mathbb{P}\left[\sup_{x \in \mathbb{R}} |G^*_\nu(x) - G(x)| > \varepsilon\right] = 0. \tag{4.12}$$

Defining $Z_\nu := \sqrt{\nu}(\hat{\theta}_\nu - \theta_0)$ the asymptotic normality assures that

$$Z_\nu \rightarrow_D Z$$

where $\mathcal{L}_{\theta_0}(Z) = G = \mathcal{N}(0,\Sigma^{-1})$. The almost-sure-representation theorem (see for instance Pollard (1989)) states that there exist $\tilde{Z}_\nu =_d Z_\nu$ and a $\tilde{Z} =_d Z$ such that

$$\tilde{Z}_\nu \rightarrow_{a.s.} \tilde{Z}.$$

As (for fixed ν) θ_0 and $\sqrt{\nu}$ are constants, Z_ν is only a function of $\hat{\theta}_\nu$. So, we can write

$$\tilde{Z}_\nu := \sqrt{\nu}(\tilde{\theta}_\nu - \theta_0)$$

for some $\tilde{\theta}_\nu \in \mathbb{R}^p$, that is

$$\tilde{\theta}_\nu := \theta_0 + \nu^{-1/2}\tilde{Z}_\nu = \theta_0 + \nu^{-1/2}\tilde{Z} + o(\nu^{-1/2}), \quad \text{a.s.}$$

Now the S-regularity of $\hat{\theta}_\nu$ gives under $P(\tilde{\theta}_\nu)$:

$$\sqrt{\nu}\left[\hat{\theta}_\nu - \tilde{\theta}_\nu\right] \rightarrow_D G, \quad \text{a.s.}$$

or in other words

$$\tilde{G}_\nu^* := \mathcal{L}_{\tilde{\theta}_\nu}\left[\sqrt{\nu}(\hat{\theta}_\nu^* - \tilde{\theta}_\nu)\right].$$

But this is to say

$$\sup_{x \in \mathbb{R}} \mid \tilde{G}_\nu^*(x) - G(x) \mid \rightarrow_{a.s} 0,$$

which implies

$$\sup_{x \in \mathbb{R}} \mid \tilde{G}_\nu^*(x) - G(x) \mid \rightarrow_{\mathbb{P}} 0. \tag{4.13}$$

Because $\tilde{G}_\nu$ is a function of $\tilde{\theta}_\nu$ (or equivalently $\tilde{Z}_\nu$) only, and because $\tilde{Z}_\nu =_d Z_\nu$, we have $\tilde{G}_\nu^* =_d G_\nu^*$ and can conclude from (4.13) that

$$\sup_{x \in \mathbb{R}} \mid G_\nu^*(x) - G(x) \mid \rightarrow_{\mathbb{P}_{\theta_0}} 0.$$

So we have derived (4.12) and Theorem 4.2 is proved completely. □

Remark 4.2 The result of Theorem 4.2 also holds for studentised versions of the parametric bootstrap. □

4.3 Description of the simulation experiment

In this section we will describe the simulation of a counting process according to the model of Jelinski-Moranda. For a description of the Jelinski-Moranda model itself we refer to Example 4.1. In our experiments we will keep the exposure period τ:=1, the sample-size K_0:=10000 and one of the model parameters, namely the true occurrence rate ϕ_0:=1 fixed. In the simulations we will vary the other model parameter N_0, the number of faults initially present in the software: $N_0 \in \{ 50,500,5000 \}$. These three cases should represent very small, normal and very large computer programs.

Remark 4.3 The reader should notice that in all formulas ϕ occurs only via $\phi\tau$. Choosing ϕ=1, τ=1 is hence in some sense not two constraints but only one; the other is just a rescaling of time-axis of the counting process $n(t)$. For all finite, positive values of ϕ and τ we have that $n(\tau)/N$ converges to $1-\exp(-\phi\tau)$, a constant fraction. In order to ensure that $n(\tau)/N$ is not too close to zero or one, as is mentioned earlier, it is appropriate to choose the stopping time τ such that the product $\phi\tau$ is about one, say $0.5 < \phi\tau < 2$. In simulations it is therefore convenient to choose ϕ=1, τ=1, although the choice ϕ=0.01, τ=100 or ϕ=100, τ=0.01 would give asymptotically identical results. □

We define for $i=1,2,...,N_0$:

$$\lambda_i := \phi_0(N_0-i+1) = N_0-i+1,$$

$$F_i(t) := 1-\mathrm{e}^{-\lambda_i t}, \ t\in[0,\tau].$$

We simulate interfailure times $t_1 \sim F_1$, $t_2 \sim F_2$,..., until

$$\sum_{i=1}^{m} t_i < \tau \quad \text{and} \quad \sum_{i=1}^{m+1} t_i \geq \tau.$$

Then we reset

$$t_{m+1} := \tau - \sum_{i=1}^{m} t_i.$$

It will turn out that for the JM-model the characteristic quantities of each simulation are $n = n(\tau)$, the number of faults detected up to time τ, and $c = c(n)$, given by

$$c(n) := \frac{1}{\tau} \sum_{i=1}^{n(\tau)+1} (i-1)t_i, \tag{4.14}$$

which is a measure for the change of the occurrence rate in time. Since $n(\tau)$ and $c(n)$ are sufficient statistics, we can spare a lot of memory space during sampling. Repeating the above described simulation K_0 times, yields two K_0-vectors: $\vec{n}:=(n_1..n_{K_0})$ and $\vec{c}:=(c_1..c_{K_0})$.

4.4 Calculation of the MLE's

We restate that in the Jelinski-Moranda model the interfailure times are assumed to be independent and exponentially distributed with parameter λ_i given by (4.1). Hence the joint pdf of the interfailure times $t_1, t_2, \ldots, t_{n(\tau)}, t_{n(\tau)+1}$ is given by (see Aalen (1978), Andersen *et al.* (1993))

$$L(N_0,\phi_0 \mid t_1,\ldots,t_{n(\tau)+1}) = \left[\prod_{i=1}^{n(\tau)} f_i(t_i)\right]\left[1-F_{n(\tau)+1}(t_{n(\tau)+1})\right] = \left[\prod_{i=1}^{n(\tau)} \lambda_i\right]\exp\left(-\sum_{i=1}^{n(\tau)+1} \lambda_i t_i\right).$$

With (4.1), the log-likelihood function is

$$\log L(N,\phi) = \sum_{i=1}^{n(\tau)} \log\phi(N-i+1) - \sum_{i=1}^{n(\tau)+1} \phi(N-i+1)t_i. \tag{4.15}$$

Hence the likelihood equations become

$$\frac{\partial}{\partial N}\log L(\hat{N},\hat{\phi}) = \sum_{i=1}^{n(\tau)} \frac{1}{\hat{N}-i+1} - \hat{\phi}\tau = 0, \tag{4.16}$$

$$\frac{\partial}{\partial \phi}\log L(\hat{N},\hat{\phi}) = \frac{n(\tau)}{\hat{\phi}} - \sum_{i=1}^{n(\tau)+1} (\hat{N}-i+1)t_i = 0. \tag{4.17}$$

With $c=c(n)$ defined by (4.14), we get from (4.15)–(4.17):

$$g(\hat{N}) := \sum_{i=1}^{n(\tau)} \frac{1}{\hat{N}-i+1}, \tag{4.18}$$

$$\hat{\phi} = \frac{n(\tau)}{(\hat{N}-c(n))\tau}. \tag{4.19}$$

It is easy to see that $n(\tau)$ and $c(n)$ are sufficient statistics in the statistical sense that we can calculate all likelihood based quantities from $n(\tau)$ and $c(n)$ only. Given the numbers $n(\tau)$ and $c(n)$ as the outcomes of a simulation (or as real life data), we can solve

$$g(\hat{N}) = 0 \tag{4.20}$$

numerically and then compute $\hat{\phi}$ with (19). Moek (1983) showed that there exists a unique solution of (4.20) if and only if

$$c(n) > \frac{n(\tau)-1}{2}. \tag{4.21}$$

To make this intuitively more likely, note that for large values of N, $g(N)$ can be approximated by

$$\begin{aligned} g(N) &= \sum_{i=1}^{n(\tau)} \left[\frac{1}{N-i+1} - \frac{1}{N-c(n)} \right] \\ &= \sum_{i=1}^{n(\tau)} \left[\frac{(N-c(n)) - (N-i+1)}{(N-i+1)(N-c(n))} \right] \\ &\sim \frac{1}{N^2} \sum_{i=1}^{n(\tau)} [i-1-c(n)] \\ &= \frac{n(\tau)}{N^2} \left[\frac{n(\tau)-1}{2} - c(n) \right]. \end{aligned}$$

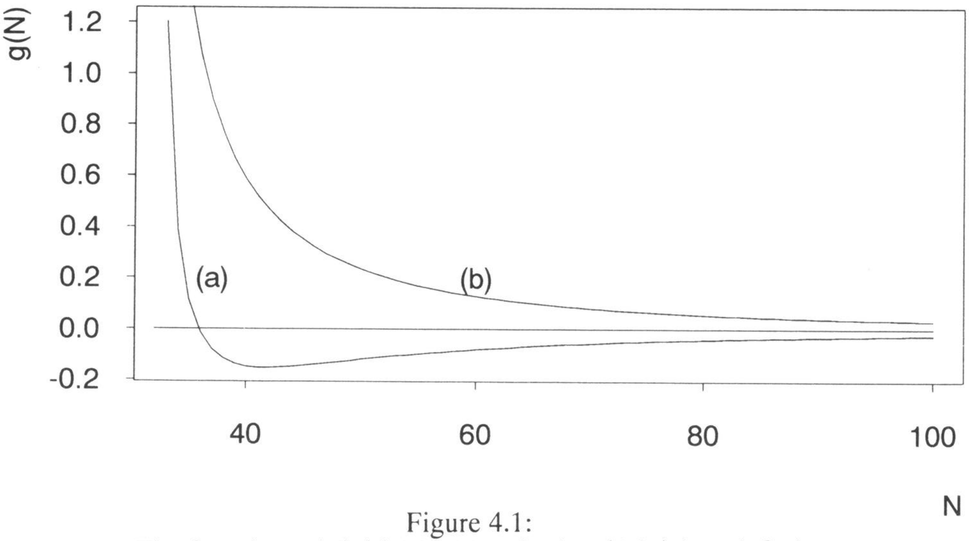


Figure 4.1:
The function $g(N)$ (a) Moek's criterion (4.21) is satisfied and (b) Moek's criterion is not satisfied.

In Figure 4.1 we have illustrated this criterion. Given these simulated vectors $\vec{n}$ and $\vec{c}$, we restrict ourselves to those simulations for which Moek's criterion is satisfied, which induces the existence and uniqueness of the solution of the ML-equations (4.16)–(4.17). We define

$$K := \#\left\{ i : 2c_i - n_i - 1 > 0 \right\}.$$

We now solve $g(\hat{N}) = 0$ (see (18)), with a variant of binary search. Then we compute the estimator $\hat{\phi}$ with (4.19). Hence this procedure yields two K-vectors of ML-estimators: $\hat{N} = (\hat{N}_1, \ldots, \hat{N}_K)$ and $\hat{\phi} = (\hat{\phi}_1, \ldots, \hat{\phi}_K)$. Applying Theorem 4.1 on the JM-model yields that we can define centered and normalised quantities $\hat{X}$ and $\hat{Y}$, satisfying

$$\begin{pmatrix} \hat{X} \\ \hat{Y} \end{pmatrix} := \begin{pmatrix} \dfrac{\hat{N} - N_0}{\sqrt{N_0}} \\ \sqrt{N_0}\,(\hat{\phi} - \phi_0) \end{pmatrix} \rightarrow_D \mathcal{N}\left(0, \begin{bmatrix} \sigma_X^2 & \tau \\ \tau & \sigma_Y^2 \end{bmatrix} \right), \tag{4.22}$$

as $N_0 \rightarrow \infty$. In (4.22), the variances σ_X^2 and σ_Y^2 are given by respectively:

$$\begin{aligned} \sigma_X^2 &:= \frac{1 - e^{-\phi_0 \tau}}{e^{\phi_0 \tau} + e^{-\phi_0 \tau} - \phi_0^2 \tau^2 - 2}, \\ \sigma_Y^2 &:= \frac{\phi_0^2 (e^{\phi_0 \tau} - 1)}{e^{\phi_0 \tau} + e^{-\phi_0 \tau} - \phi_0^2 \tau^2 - 2}. \end{aligned} \tag{4.23}$$

This follows directly from (4.5).

4.5 Construction of confidence intervals

We will construct upper confidence bounds and two-sided confidence intervals for N, based on the ML-estimator $\hat{N}$. For this purpose, we will make use of an approximate normal test statistic z and of a likelihood ratio test statistic w. From (4.22) it follows directly that we have approximately

$$z := \frac{\hat{N} - N_0}{\sigma \sqrt{\hat{N}}} \sim \mathcal{N}(0, 1),$$

where the asymptotic variance σ^2 is given by (4.23). We can estimate σ consistently by using either the expected information matrix Σ (see (4.5)), which yields

$$\hat{\sigma}_{EXP} := \left[\frac{1-e^{-\hat{\phi}\tau}}{e^{\hat{\phi}\tau}+e^{-\hat{\phi}\tau}-\hat{\phi}^2\tau^2-2} \right]^{1/2}$$

$$= \left[(e^{\hat{\phi}\tau}-1) - \frac{\hat{\phi}^2\tau^2}{1-e^{-\hat{\phi}\tau}} \right]^{-1/2}, \tag{4.24}$$

or by using the observed information matrix I (see (4.10)), which yields

$$\hat{\sigma}_{OBS} := \left[\hat{N} \sum_{i=1}^{n(\tau)} \frac{1}{(\hat{N}-i+1)^2} - \frac{\hat{N}\hat{\phi}^2\tau^2}{n(\tau)} \right]^{-1/2}.$$

As it turned out that results based on $\hat{\sigma}_{OBS}$ did not differ significantly from those based on $\hat{\sigma}_{EXP}$, we will in Section 4.4 only discuss the latter, which are easier to compute.

Let k_α and $k_{\alpha/2}$, $\alpha \in [0, \frac{1}{2}]$, be quantiles of the standard normal distribution, such that

$$\Pr(|z| \le k_\alpha) = 1-\alpha,$$

$$\Pr(|z| \le k_{\alpha/2}) = 1-\alpha/2.$$

If $\hat{\sigma}$ denotes either $\hat{\sigma}_{EXP}$ or $\hat{\sigma}_{OBS}$ an approximate $(1-\alpha)$-upper-bound U_{normal} is given by

$$U_{\text{normal}} := \hat{N}+k_\alpha \hat{\sigma}\sqrt{\hat{N}}$$

and $(LB_{\text{normal}}, UB_{\text{normal}})$ is an approximate $(1-\alpha)$ confidence interval where

$$LB_{\text{normal}} := \hat{N}-k_{\alpha/2}\hat{\sigma}\sqrt{\hat{N}},$$

$$UB_{\text{normal}} := \hat{N}+k_{\alpha/2}\hat{\sigma}\sqrt{\hat{N}}.$$

Another way of constructing upper-bounds and confidence intervals for $\hat{N}$ can be done by making use of the Wilks likelihood ratio test (WLRT) statistic

$$w := 2[\max_{\phi} \log L(\hat{N},\phi) - \max_{\phi} \log L(N_0,\phi)],$$

where the log-likelihood function $\log L$ is given by (4.4). As indicated in Van Pul (1990), the WLRT statistic w is asymptotically chi-squared distributed with one degree of freedom. Let c_α and $c_{2\alpha}$ denote the $(1-\alpha)$ and $(1-2\alpha)$ quantiles of the $\chi^2(1)$-distribution. Then an alternative $(1-\alpha)$-upper-bound U_{Wilks} is the largest solution x of

$$G(x):=2[\max_{\phi}\log L(\hat{N},\phi)-\max_{\phi}\log L(x,\phi)]=c_{\alpha}$$

and if LB_{Wilks} and UB_{Wilks} are the solutions of

$$G(x):=2[\max_{\phi}\log L(\hat{N},\phi)-\max_{\phi}\log L(x,\phi)]=c_{2\alpha},$$

with $LB_{\text{Wilks}}<\hat{N}<UB_{\text{Wilks}}$, then $(LB_{\text{Wilks}},\ UB_{\text{Wilks}})$ is an alternative $(1-\alpha)$-confidence interval for the parameter of most interest N. See Figure 4.2.

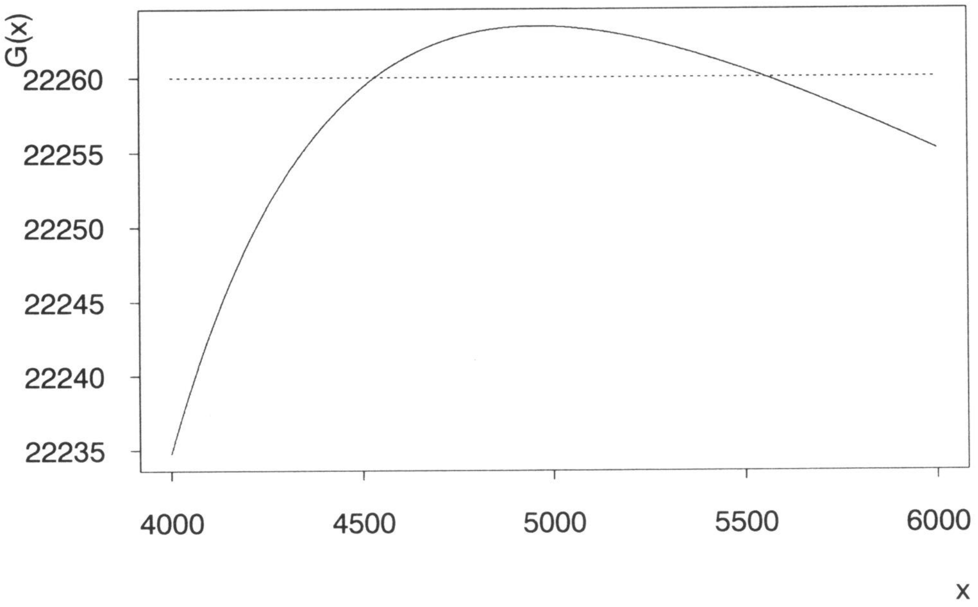


Figure 4.2:
The function G(x).

The computation of approximate normal upper confidence bounds and confidence intervals is much easier, but those calculated from the WLRT statistic will be much more satisfying. To spare computer time in the latter case we will approximate sums (with 500-5000 terms) by standard integrals with well-known primitives.

Finally, to construct bootstrapped confidence intervals we used a set consisting of 1000 ML-estimator-pairs $(\hat{N}_i,\hat{\phi}_i)$, $i=1...1000$, simulated according to the Jelinski-Moranda model with $N_0=50$ and $\phi_0=1$. For each estimator-pair $(\hat{N}_i,\hat{\phi}_i)$ we now constructed 1000 bootstrapped estimator-pairs $(\hat{N}^*_{ij},\hat{\phi}^*_{ij})$, $j=1...1000$ and used formulas analogous to (4.9) to construct one- and two-sided confidence intervals for N_0.

The empirical coverage probability or hitting percentage (hit%) of a confidence interval is defined to be the fraction of the K constructed confidence intervals, which contain the true number of faults N_0. The miss-under percentage (mu%) is the fraction of confidence intervals whose upper-bound is lower than N_0. In an analogous manner we define the miss-over percentage (mo%).

4.6 Discussion of the results of the experiment

Once again, we mention the fact that during the simulation experiment we will keep the exposure period τ=1 and the true occurrence rate ϕ_0=1 fixed. As we have studied some theoretical properties of the ML-estimators as $N_0 \rightarrow \infty$ in Van Pul (1990), we are very interested for which size of N_0 this asymptotic behaviour appears in practice. Therefore, we do the simulations and computations, described in the last section for various values of N_0 in the range from 50 to 5000. Simulations with values of N_0, essentially bigger then 5000, were not possible (and would not be interesting) with the available computing facilities. For each value of N_0, we repeat the experiment K_0=10000 times. [The same simulated data were already briefly analysed in Example 3.3 in Section 3.5. See therefore also Tables 3.3-3.4 and Figures 3.3-3.4.]

In Table 4.1 we give the number of the finite estimators. That is, the number of estimators which corresponding simulation characteristics $n(\tau)$ and $c(n)$, satisfy Moek's criterion (4.21). In fact this number K is the realised number of replicates. In Table 4.2 means and mean square errors for the ML (point-) estimators $\hat{N}$ and $\hat{\phi}$ are given. In Table 3.3 means and variances of the centered and normalised quantities $\hat{X}$ and $\hat{Y}$ are given. In Table 4.3 we compared the approximate normal and Wilks likelihood ratio upper confidence bound for N_0 at various values of the confidence level p:=100(1−α). Using the same simulation data we computed the realised hitting percentages. In Table 4.4 we did the same for two-sided confidence intervals derived from the approximate normal and the Wilks ratio likelihood test statistics.

N_0	K
50	9027
500	9996
5000	10000

Table 4.1
Number of finite estimators.

For small values of N_0, we see in Table 4.1 that $K < K_0$. When N_0 increases, K/K_0 converges quickly to 1. For small values of N_0 the variation in n can be relatively large. So possibly some simulations will exist that don't satisfy Moek's criterion: $2c-n-1>0$. On the other hand we have as $N_0 \rightarrow \infty$:

$$\frac{n}{N_0} \rightarrow_{\mathrm{IP}} x_0(\tau) = 1-e^{-1},$$

$$\frac{c}{N_0} = \frac{1}{N_0}\int_0^\tau n(s-)\mathrm{d}s \rightarrow_{\mathrm{IP}} \mathrm{e}^{-1}.$$

Hence $n/c \rightarrow_{\mathrm{IP}} \mathrm{e}-1$ as $N_0 \rightarrow \infty$ and Moek's criterion will be satisfied with a probability tending to one, when N_0 grows larger.

N_0	$\mathrm{IE}\hat{N}$	$\mathrm{IE}\hat{\phi}$	$\mathrm{IE}(\hat{N}/N_0-1)^2$	$\mathrm{IE}(\hat{\phi}-\phi_0)^2$
50	48.3947	1.3095	0.0909	0.3982
500	506.1666	1.0228	0.0194	0.0406
5000	5008.0997	1.0015	0.0015	0.0039

Table 4.2
Means and mean square errors of the MLE's $\hat{N}$ and $\hat{\phi}$.

For small values of N_0 we see in Table 4.2, that $\hat{N}\ll N_0$ and $\hat{\phi}\gg\phi_0$. As N_0 increases, we find however, by looking at the mean square errors, confirmation of the theoretical results:

$$\frac{\hat{N}}{N_0} \rightarrow_{\mathrm{IP}} 1 \quad \text{and} \quad \hat{\phi} \rightarrow_{\mathrm{IP}} \phi_0 = 1,$$

derived in Van Pul (1990). The under-estimation of N_0 can be explained by the fact that for small values of N_0, K is smaller than K_0 and the biggest estimates for N_0 are omitted. In the same way the over-estimation of ϕ_0 by $\hat{\phi}$ can be explained, because $\hat{\phi}\sim 1/\hat{N}$ (see (4.19)).

The distribution of $\hat{N}$ is even for large values of N_0 not symmetric at all, but has a very long tail on the right. The skewness of the distribution of $\hat{N}$ is related to stability problems of the numerical approximation procedure. A vertical perturbation of the curve of the likelihood function would only cause a small deviation of the solution $\hat{N}$ to the left, but easily cause a large one to the right. See Figure 4.2. This skewness disappears slowly as N_0 increases.

We see in Table 3.3 that the convergence of $\hat{X}$ and $\hat{Y}$ to normal distributions with means zero and asymptotic variances as expected (in (4.22)) is rather slow. The difference in the asymptotic behaviour of $\hat{N}$ and $\hat{\phi}$ is illustrated by the histograms and qq-plots given in Figures 3.3 and 3.4. Both tables and figures give the same impression, namely that the distribution of $\hat{N}$ shows a severe skewness and that the distribution of $\hat{\phi}$ is rather biased for small N_0. Both defects slowly disappear as N_0 increases.

We will now consider the construction of confidence intervals for N_0. For the construction of confidence intervals based on the approximate normal test statistic we estimated the asymptotic variance with use of the expected information (see (4.24)). If the observed information had been used, the results would not have been significantly different. As the distribution of $\hat{N}$ is skew, however, the coverage percentages of confidence intervals based on the asymptotic normal statistic could be expected to be disappointing.

In Table 4.3 we compare these percentages with those based on the Wilks test statistic (4.6). We see that at the 50% confidence level the approximate normal and the Wilks likelihood ratio upper-bounds both are equal to the maximum likelihood estimator $\hat{N}$. For higher confidence levels p, the Wilks upper confidence bound is larger than the corresponding approximate normal one. The same will then of course hold for the coverage probabilities. All these phenomena are clearly understood by the way of constructing the upper-bounds. As the Wilks confidence intervals are larger, shifted to the right (and hence not symmetric around N_0) in comparison with the approximate normal confidence intervals, for high levels of confidence the Wilks intervals are significantly better and have less skew coverage probabilities than the approximate normal ones. Note that the number of replicates K_0=10000.

Finally, we consider two-sided confidence intervals. In Table 4.4 we observe a skewness in the hitting probabilities, which appears to be significantly larger in case of the approximate normal confidence intervals for N_0. This skewness decreases with increasing confidence level p and with increasing N_0. The size of an approximate normal confidence interval, which is by definition symmetric around $\hat{N}$, increases of course with increasing p, but it is also proportional to $\hat{N}$. This explains the extreme skewness of the hitting probabilities of the approximate normal confidence intervals. Wilks confidence intervals behave better for two reasons: they are bigger and shifted to the right in comparison with the approximate normal confidence intervals. In Table 4.4 we can see that for high levels of confidence the Wilks confidence intervals are significantly better and less skew than the approximate normal ones.

Recently we made the beginning with a study of the behaviour of the ML-estimators in practice, computed from data simulated according to the Jelinski-Moranda model. We compared the estimated coverage probabilities of confidence intervals for $\hat{N}$ based on the approximate normal and on the Wilks likelihood ratio test statistic. From the simulation results, presented by Van Pul (1991a), we can draw several conclusions. Firstly, they confirm the asymptotic theory as derived in Van Pul (1990) for large N_0 (~5000). For medium large N_0 (~500) the simulation results show that the asymptotic theory does not occur yet, but also shows that the Wilks likelihood ratio test statistic provides confidence intervals which are much better.

(a)	normal		Wilks	
p	hit%	$\mathbb{E}(U)$	hit%	$\mathbb{E}(U)$
50	34	48	35	49
55	39	50	38	51
60	44	53	44	55
65	48	56	49	60
70	52	59	55	72
75	56	62	61	∞
80	60	65	68	∞
85	64	69	75	∞
90	69	74	82	∞
95	75	82	90	∞

(b)	normal		Wilks	
p	hit%	$\mathbb{E}(U)$	hit%	$\mathbb{E}(U)$
50	46	506	46	507
55	51	514	51	514
60	56	523	56	523
65	60	532	61	534
70	65	541	66	545
75	69	551	71	558
80	73	562	76	575
85	78	575	82	596
90	83	591	88	627
95	88	615	94	685

(c)	normal		Wilks	
p	hit%	$\mathbb{E}(U)$	hit%	$\mathbb{E}(U)$
50	49	5008	49	5009
55	54	5032	54	5032
60	59	5056	59	5057
65	64	5082	64	5084
70	69	5109	69	5112
75	74	5138	75	5144
80	79	5170	80	5180
85	84	5208	85	5222
90	88	5255	90	5278
95	93	5326	95	5364

Table 4.3
Hitting percentages and mean upper-bounds of one-sided confidence intervals
(a) N_0=50, (b) N_0=500, (c) N_0=5000.

(a)	normal					Wilks				
p	mu%	hit%	mo%	IE(LB)	IE(UB)	mu%	hit%	mo%	IE(LB)	IE(UB)
50	44	56	0	34	62	38	53	9	39	∞
55	42	58	0	32	64	36	57	7	39	∞
60	40	60	0	30	65	32	62	6	38	∞
65	38	62	0	29	67	29	67	4	37	∞
70	36	64	0	26	69	25	72	3	37	∞
75	34	66	0	24	72	22	76	2	36	∞
80	31	69	0	21	74	18	81	1	36	∞
85	28	72	0	18	78	14	85	1	35	∞
90	25	75	0	14	82	10	90	0	34	∞
95	22	78	0	7	88	6	94	0	33	∞

(b)	normal					Wilks				
p	mu%	hit%	mo%	IE(LB)	IE(UB)	mu%	hit%	mo%	IE(LB)	IE(UB)
50	31	51	18	461	551	28	50	22	466	588
55	29	56	15	456	556	26	55	19	462	566
60	27	61	12	450	562	23	60	17	458	575
65	25	66	9	444	568	20	65	15	454	585
70	22	72	6	437	575	18	70	12	449	596
75	20	77	3	430	582	15	75	10	445	610
80	17	82	1	421	591	12	80	8	439	627
85	14	86	0	410	602	9	85	6	433	650
90	12	88	0	397	615	6	90	4	426	685
95	9	91	0	376	636	3	95	2	415	780

(c)	normal					Wilks				
p	mu%	hit%	mo%	IE(LB)	IE(UB)	mu%	hit%	mo%	IE(LB)	IE(UB)
50	27	50	23	4877	5138	26	50	24	4883	5143
55	24	44	21	4862	5154	23	55	22	4869	5160
60	21	61	18	4845	5170	20	60	20	4854	5179
65	19	66	15	4827	5188	18	65	17	4837	5199
70	16	71	13	4807	5208	15	71	14	4820	5221
75	14	76	10	4785	5230	12	76	12	4801	5247
80	12	81	7	4760	5255	10	80	10	4779	5247
85	9	86	5	4729	5286	8	85	7	4753	5314
90	7	90	3	4690	5326	5	90	5	4721	5363
95	4	95	1	4629	5386	3	95	2	4672	5441

Table 4.4
Hitting/miss-percentages and mean lower/upper-bounds of two-sided confidence intervals
(a) N_0=50, (b) N_0=500, (c) N_0=5000.

Finally, for small N_0 (~50) we found that confidence intervals based on the WLRT statistic still have reasonable estimated coverage probabilities; the upper-bounds of those confidence intervals can be extremely large and sometimes even be infinite, however. There is a clear interpretation of the model and its likelihood when $N_0=\infty$; we are then namely in the special case of a Poisson process with constant intensity. In this section we will therefore concentrate on the case that N_0 is small (say 50) and compare estimators obtained from parametric bootstrapping methods with the estimators (based on the approximate normal and on the WLRT test statistic) constructed earlier.

p	50	55	60	65	70	75	80	85	90	95
AN	35	39	44	47	51	55	59	63	69	75
W	37	39	44	49	55	62*	68*	75*	81*	89*
B	44	46	48	51	53	57	58	63	67	71
SB	47	51	55	62	66	72	76	80	86	90

Table 4.5
Hitting percentages of one-sided confidence intervals.

p	50	55	60	65	70
AN	45:55:0	43:57:0	41:59:0	38:62:0	37:63:0
W	39:52:9*	35:58:7*	32:63:5*	28:68:4*	25:72:3*
B	43:31:26	42:34:24	41:37:22	40:40:20	38:45:17
SB	28:45:27	27:47:26	25:52:23	22:56:22	21:58:21

p	75	80	85	90	95
AN	34:66:0	31:69:0	28:72:0	25:75:0	22:78:0
W	22:76:2*	19:80:1*	14:85:1*	11:89:0*	7:93:0*
B	36:49:15	34:53:13	33:56:11	30:61:9	26:68:6
B	18:62:20	16:66:18	12:71:17	10:75:15	7:81:12

Table 4.6
Hitting and miss percentages of two-sided confidence intervals.

In Table 4.5 we compared the estimated coverage probabilities of upper-bounds (for various levels of confidence) constructed with use of the approximate normal test statistic (AN), the Wilks likelihood ratio test statistic (WLR), (non-studentised) bootstrapping (B) and studentised bootstrapping (SB). An asterisk (*) denotes that upper-bounds were sometimes infinite.

Analogously in Table 4.6 hitting and miss percentages of two-sided confidence intervals are given. In each entry $(x_1{:}x_2{:}x_3)$ in Table 4.6, x_1 corresponds with the miss-under percentage (that is the percentage of the confidence intervals with upper-bound smaller than N_0), x_2 denotes the hitting percentage and x_3 represents the miss-over percentage. Again an asterisk (*) denotes that upper-bounds were sometimes infinite.

From Table 4.5 and Table 4.6 we may conclude the following:

(a) Non-studentised bootstrap results are poor; studentising has proved to be very fruitful.

(b) Hitting percentages of one-sided confidence intervals (upper-bounds) based on a studentised bootstrap method, are comparable to those based on the WLRT statistic (and much better than those based on the approximate normal test statistic).

(c) Hitting percentages of two-sided confidence intervals based on a studentised bootstrap method, are comparable to those based on the approximate normal test statistic (and significantly worse than those based on the WLRT statistic).

(d) (Studentised) bootstrap upper-bounds are, in contrary to the WLRT ones, always finite.

(e) (Studentised) bootstrap results have the advantage (in comparison with the approximate normal and WLRT results) that the miss percentages are more symmetrically distributed.

Except perhaps for (c), all observations confirm the developed theory. It is not clearly understood, however, why the studentised bootstrap results for two-sided confidence intervals are less promising than expected. Apparently, studentising causes a downswing in the hitting percentages of the lower-bounds in our case.

4.7 Concluding remarks

In this chapter we have only made a beginning with the study of the behaviour of the ML-estimators (in practice), computed from simulated data. As stated before, it was not our intention to prove or disprove the validity of the Jelinski-Moranda model for real data sets. We wanted to get some more insight in the asymptotic behaviour of the ML-estimators, assuming the model is correct. The simulation results discussed in the previous section, confirm the consistency and asymptotic normality of the maximum likelihood estimators (as derived in Van Pul (1990)), but also show that asymptotic convergence in distribution is appearing very slowly and that for small values of N_0 the distribution of $\hat{N}$ can be very skew. On the other hand, asymptotic confidence intervals, constructed with the Wilks likelihood test statistic, have coverages probabilities close to the corresponding confidence levels, even at low sample-sizes.

For N_0=50 we have further modified the estimated coverage probabilities for two-sided Wilks confidence intervals by taking into account also data-points $(n(\tau), c(n))$ where Moek's criterion is not satisfied. This corresponds with $\hat{N}=\infty$ (see Figure 4.2). Note that the Jelinski-Moranda model has as a special case the Poisson model (with constant failure intensity λ_0); this is the limit-case letting $N_0 \rightarrow \infty$ and $\phi_0 \rightarrow 0$ such that $\lambda_0 := \phi_0 N_0$ is a constant. We can easily derive and maximise the (log) likelihood function for this special case. We have constructed confidence intervals of the form (lower-bound, infinity) and considered them as two-sided confidence intervals for N_0. In Table 4.7 we give the original values P_{old} (see Table 4.4), correction terms $P_{infinite}$ and the corrected values P_{new} for the estimation of the coverage probabilities. We computed P_{new} from

$$P_{new} = \frac{K}{K_0} P_{old} + (1-\frac{K}{K_0}) P_{infinite},$$

where K=9027 and K_0=10000 are respectively the number of finite estimators (see Table 4.1) and the sample-size. The figures of Table 4.7 show that the new hitting percentages are a little bit worse than the old ones; the miss-rates mu% and mo%, however, are more symmetrically distributed. Probably the old values P_{old} (totally ignoring infinite estimators $\hat{N}$) were (a bit) too optimistic.

	P_{old}			$P_{infinite}$			P_{new}		
p	mu%	hit%	mo%	mu%	hit%	mo%	mu%	hit%	mo%
50	39	52	9	0	15	85	35	49	16
55	36	57	7	0	17	83	32	54	14
60	32	62	6	0	21	79	29	58	13
65	29	67	4	0	27	73	26	63	11
70	25	72	3	0	33	67	23	68	9
75	22	76	2	0	42	59	20	73	7
80	18	81	1	0	49	51	16	78	6
85	14	85	1	0	61	39	13	83	4
90	10	90	0	0	72	28	9	88	3
95	6	94	0	0	89	11	5	94	1

Table 4.7
Corrected hitting and miss percentages of
(modified) two-sided Wilks confidence intervals for N_0=50.

We plan to investigate various ways of improving the asymptotic approximation. In Van Pul (1991c) we studied the asymptotic behaviour of parametric *bootstrap methods.* We have proved that the parametric bootstrap (in the context of counting processes) works, that is, is asymptotically consistent. Computer simulations showed that with a studentised version of the parametric bootstrap we can construct one-sided confidence intervals with hitting percentages even better than those constructed with Wilks LRT statistic. For two-sided confidenced intervals, constructed with a studentised version of the parametric bootstrap, we obtained less satisfying figures. We are trying to improve this. With use of *Edgeworth expansions* we can probably determine the second order term of the limit distribution of $\hat{N}$.

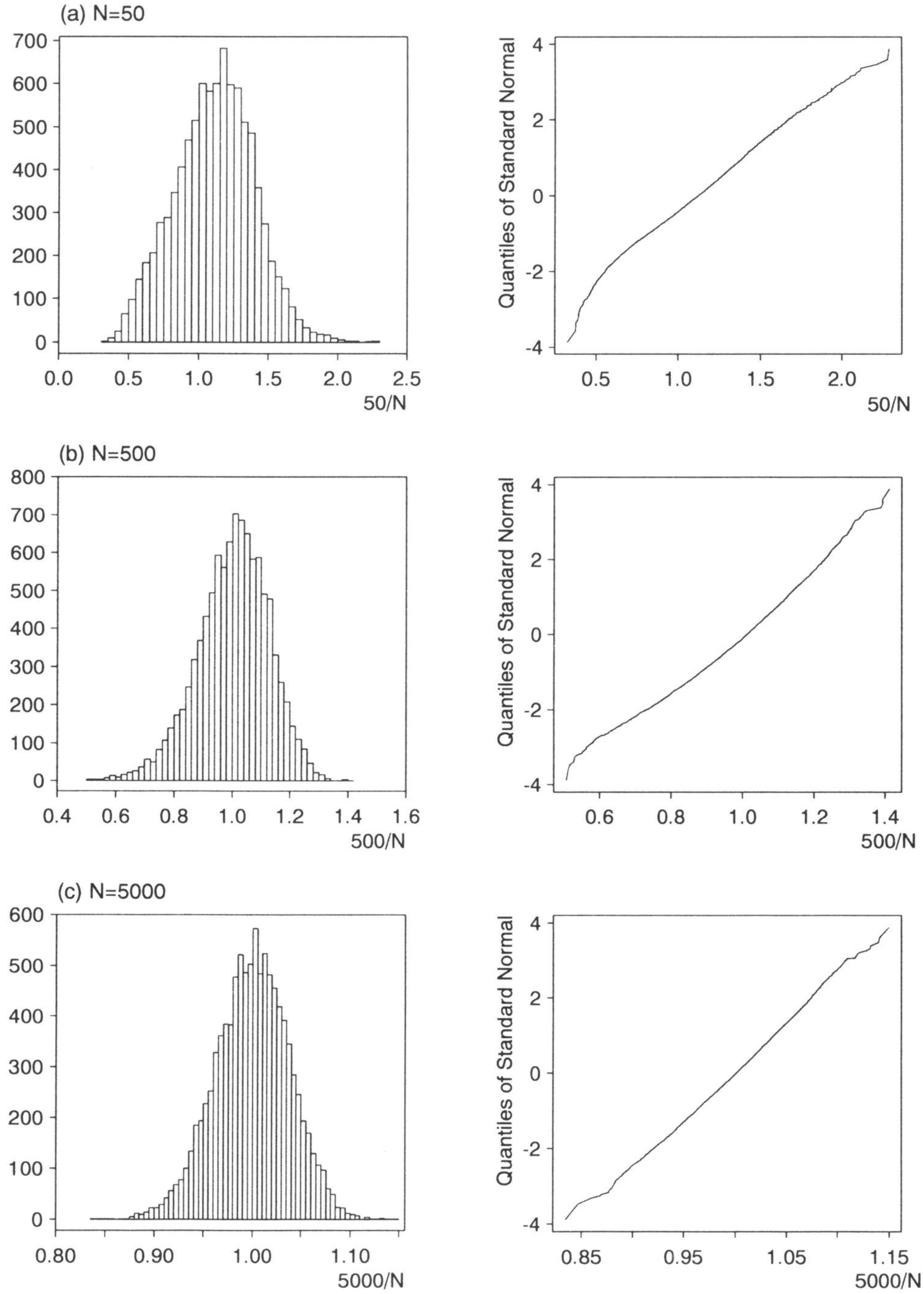


Figure 4.3:
Histograms and qq-plots of $1/\hat{N}$.
(a) N_0=50 (b) N_0=500 (c) N_0=5000

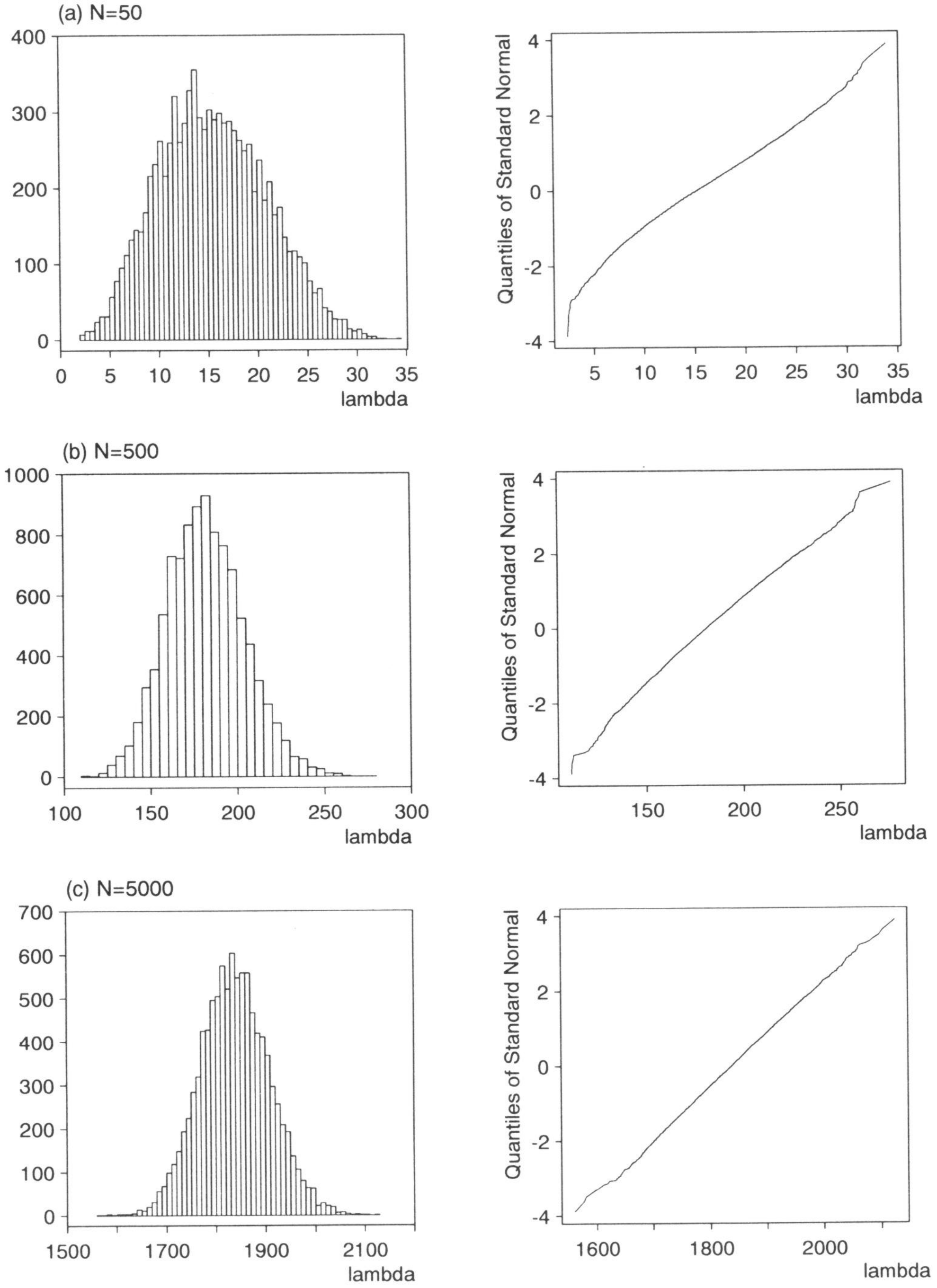


Figure 4.4:
Histograms and qq-plots of $\hat{\lambda}$
(a) N_0=50 (b) N_0=500 (c) N_0=5000

Another direction for further research seems to be the investigation of *certain functions of the model parameters.* Using the delta-method (first used by Kendall & Stuart (1967), Rao (1973), Gill (1989)) we can derive immediately asymptotic results (convergence to normal distribution) for decent functions of the parameters N and ϕ. Entities like $1/\hat{N}$ and $\hat{\lambda}:=\hat{\phi}(\hat{N}-n)$ behave already for small values of N_0 remarkably well: they have smaller variances, are less skewed in comparison with $\hat{N}$ and converge faster to normality (see Table 4.8 and Figures 4.3 and 4.4).

N_0	Var $\hat{N}$	Var $\hat{\lambda}$
50	225	183
500	4804	2028
5000	37066	20395

Table 4.8
Comparison of the variances of $\hat{N}$ and $\hat{\lambda}:=\hat{\phi}(\hat{N}-n)$.

Of course, we can use asymptotic results of for instance $1/\hat{N}$ to construct alternative confidence intervals for N. It is, however, perhaps even more interesting to study and estimate quantities like $\hat{\lambda}$ that have a clear interpretation, also when the assumed underlying model is not true.

Chapter 5

Solutions to the Consistency Problem for the Littlewood Model in Software Reliability*

5.1 Introduction

A very well-known model in software reliability theory is that of Littlewood (1980). The parameters in this model are usually estimated by means of the maximum likelihood (ML) method. The system of likelihood equations can have more than one solution. Only one of them will be consistent, however. We present and compare different approaches to construct estimators for the model parameters of this particular model and investigate whether they are consistent or not. Our belief is that the ideas and methods developed in this paper could also be of interest for statisticians working outside the field of (software) reliability theory.

Maximum likelihood estimators are generally used in all fields of statistics. For those MLE's theoretical (asymptotic) properties are often derived, but seldom verified in practice. Le Cam (1990) showed how terribly they can behave sometimes. Another thing to keep in mind is the fact that in many cases the likelihood equations will have more than one solution. Classical theorems state that under suitable conditions exactly one of those solutions of the likelihood equations will be consistent and that this consistent solution will be asymptotically normally distributed and efficient. Here the problem arises how to

* This chapter is based on CWI report BS-R9129 and on the paper "Solutions to the Consistency Problem for the Littlewood Model in Software Reliability" by Leo G. Barendregt and Mark C. van Pul, which is submitted for possible publication to *Statistica Neerlandica.*

choose from a couple of candidates (solutions of the likelihood equations) the consistent one. More seriously, if one has laboriously numerically determined one solution, how can one be sure there are no others? Le Cam (1990) addresses those kind of problems too. His advice is to just apply one-step of the Newton-Raphson method to an initial estimator, which is $\sqrt{n}$-consistent. It is a well-known result that such an one-step estimator will be asymptotically equivalent to the MLE. Practical results obtained with the one-step Newton-Raphson method, however, often turn out to be very disappointing. Moreover, it is often actually rather difficult to construct a suitable initial estimator.

In this paper we consider this problem for a particular case in the field of software reliability theory. In the next section we introduce Littlewood's parametric software reliability model in more detail. In Section 5.3 we discuss the maximum likelihood estimation method for counting processes and show that for the software reliability model of Littlewood (1980) the likelihood equations can have more than one solution. Hence we face here the problem mentioned earlier. In Sections 5.4 and 5.5 a first approach is presented how an initial estimator, that is $\sqrt{n}$-consistent, can be found. We describe an algorithm based on this that produces an asymptotically efficient estimator. In practice this algorithm works well (and better than the one-step Newton-Raphson method). In Sections 5.6 and 5.7 we describe a second, more analytical approach to the problem, exploiting the mathematical properties of the log-likelihood function of the Littlewood model. We will prove that the global maximiser of the likelihood is consistent, and discuss how to find it numerically. Recently we found a third alternative way of constructing consistent estimators for the parameters of the Littlewood model. This method III, which deals with modified score equation solutions, seems more generally applicable than the two other methods and is sketched in Section 5.8. A more thorough discussion of this method will be contained in Barendregt & Van Pul (1993). Finally, in Section 5.9 we compare some numerical results of the methods I and II, when applied to simulated data, generated by the Littlewood model.

5.2 The Littlewood model

Computer systems have become more and more important in modern society. The problem of estimating the reliability of computer software undergoing debugging has therefore, over the last two decades, received a great deal of attention. For this purpose a considerable number of models has been proposed. We refer to Musa *et al.* (1987) for a complete overview of the most common software reliability models. Two of the most well-known models in software reliability theory are those of Jelinski-Moranda (1972) and Littlewood (1980). For the Jelinski-Moranda model Moek (1983) gave a criterion on the data, satisfied with probability one when the model is true, under which there exists an unique solution of the maximum likelihood equations. For the model of Littlewood such a criterion, however, is not known and probably will not exist. With use of asymptotic theory it is proved (Van Pul (1992)) that in case of one or more solutions of the likelihood equations exactly one of them will be consistent, if we can make this choice using the data only. Here the earlier mentioned problem arises: how to choose a consistent MLE, when the likelihood equations have more than one solution. Also alternative estimation methods do have the same difficulties with the Littlewood model. In case of M-estimation, nonsingularity of the resulting matrix has unfortunately not been proved yet (see Geurts *et al.* (1988)).

We consider the following test experiment. A computer program has been executed during a specified exposure period and the interfailure times are observed. The repairing of a fault takes place immediately after it produces a failure and no new faults are introduced with probability one. Let N be the unknown number of faults initially present in the software. Let the exposure period be $[0,\tau]$ and let $n(t)$, $t \in [0,\tau]$, denote the number of faults detected up to time t. Define $T_0 := 0$ and let T_i, $i=1,2,...,n(\tau)$, the failure time of the i-th occurring failure, while $t_i := T_i - T_{i-1}$, $i=1,2,...,n(\tau)$, denotes the interfailure time, that is the time between the i-th and the $(i-1)$-th occurring failure. Finally we define $t_{n(\tau)+1} := \tau - T_{n(\tau)}$.

In the Jelinski-Moranda model, introduced in 1972 and a few years later generalised by Musa (1975), the failure rate of the program is at any time proportional to the number of remaining faults and each fault still present makes the same contribution to the failure rate. So if $(i-1)$ faults have already been detected, the failure rate for the i-th occurring failure, λ_i, becomes

$$\lambda_i = \phi_0 \left[N_0 - (i-1) \right], \tag{5.1}$$

where ϕ_0 is the true failure rate per fault (the occurrence rate) and N_0 is the true number of faults initially present in the software. In terms of counting processes we can write

$$\lambda^{\mathrm{JM}}(t) = \phi_0 \left[N_0 - n(t-) \right], \quad t \in [0,\tau], \tag{5.2}$$

where $\lambda(t)$, $t \in [0,\tau]$ denotes the failure rate at time t. The interfailure times $t_i, i=1,...,n(\tau)$, are independent and exponentially distributed with parameter λ_i given by (5.1).

In the model, introduced by Littlewood (1980), it is again assumed that at any time the failure rate is proportional to the number of remaining errors. The main difference in the Littlewood model with respect to the Jelinski-Moranda model, is the fact that each fault does not make the same contribution to the failure rate $\lambda(t)$. Littlewood's argument for that is that larger faults will produce failures earlier than smaller ones. He treats ϕ_j, the failure rate of fault j, as a stochastic variable and suggests a Gamma distribution:

$$\phi_j \sim \Gamma(a_0, b_0), \quad j=1,...,N.$$

We define the expected occurrence rate of faults not occurred up to time t as

$$\phi(t) := \mathbb{E}(\phi_j \mid T_j > t),$$

with

$$\phi_j \;\sim\; \Gamma(a_0, b_0),$$

$$T_j \mid \phi_j = \phi \;\sim\; \exp(\phi).$$

A simple calculation yields that the T_j have excess probability

$$\Pr(T_j>t) = \left(\frac{b_0}{b_0+t}\right)^{a_0},$$

that is, the T_j have a generalised Pareto distribution. Furthermore

$$\phi_j \mid T_j>t \sim \Gamma(a_0, b_0+t)$$

and hence

$$\phi(t) = \frac{a_0}{b_0+t}.$$

An application of the so called innovation-theorem (Aalen (1978)) now shows, that the failure intensity of the software at time t is given by

$$\lambda^{L}(t)=\frac{a_0\left(N_0-n(t-)\right)}{b_0+t}. \tag{5.3}$$

By a simple reparametrisation, namely:

$$\alpha_0=\frac{a_0}{b_0}, \quad \varepsilon_0=\frac{1}{b_0},$$

we get from (5.1):

$$\lambda^{GL}(t) = \frac{\alpha_0\left(N_0-n(t-)\right)}{1+\varepsilon_0 t}, \quad t\in[0,\tau]. \tag{5.4}$$

Actually formula (5.4) provides an extension of the Littlewood model (5.3), allowing also small values of $\varepsilon_0 \le 0$. Restricting ourselves to the conventional parameter space

$$\Theta := \{ (N,\alpha,\varepsilon)\in \mathrm{IR}^3 \mid N\ge 0, \ \alpha\ge 0, \ \varepsilon\ge 0 \},$$

which is not compact, we investigate the model behaviour at the boundary of the parameter-set. We will see that letting certain combinations of N, α, ε converge to their boundary limits (zero or infinity) at various speeds, this may lead to different limiting models. Apart from the *null-model (0)* (where, as for instance $N=0$, nothing happens) we can roughly distinguish four non-trivial boundary models:

(1) *The explosion model* (E^{δ}):
If for instance $0<N<\infty$, $\alpha=\infty$ and $0<\varepsilon,\infty$, the failure-intensity at time zero becomes infinite. Thus the expected number of bugs detected makes a jump from zero to δ at time zero and remains constant for $t>0$. The failure-intensity drops to zero for $t>0$. Special cases are E^0 (the null-model), E^N and E^{∞}. E^{Δ} represents the class of all explosion models $\{E^{\delta} \mid 0\leq\delta\leq\infty\}$.

(2) *The Jelinski-Moranda model (JM):*
If $0<N<\infty$, $0<\alpha<\infty$ and $\varepsilon=0$, we are dealing with the model we discussed earlier, namely the *Jelinski-Moranda model* (5.2) with occurrence rate parameter ϕ equal to α. We can therefore treat the Jelinski-Moranda model as a special (limit-)case of the Littlewood model.

(3) *The inhomogeneous Poisson model (IP):*
If $N\rightarrow\infty$, $\alpha\rightarrow 0$ and $0<\varepsilon<\infty$ such that $N\alpha\rightarrow\delta$, the influence of the past of the counting-process $n(t-)$ is eliminated from expression (5.4) and the general model reduces to an *inhomogeneous Poisson model* with intensity function $\lambda(t) = \delta/(1+\varepsilon t)$, $t\in[0,\tau]$.

(4) *The homogeneous Poisson model (HP):*
If $N\rightarrow\infty$, $\alpha\rightarrow 0$ and $\varepsilon\rightarrow 0$ such that $N\alpha\rightarrow\delta$, all time-dependence is eliminated from the expression (5.4) and the general model reduces to a *homogeneous Poisson model* with constant failure intensity δ.

Table 5.1 shows which of the above mentioned limiting models occur for which (N,α,ε). Note that for instance for $0<N<\infty$, $\alpha\rightarrow\infty$ and $\varepsilon\rightarrow\infty$ the limiting model heavily depends on the way we let α and ε increase. In Figure 5.1 we plotted expected number of faults detected versus time for various choices of parameter triples (N,α,ε) approaching the boundary of the parameter-set Θ. The bold curves represent the limit models.

	$N=0$			$0<N<\infty$			$N=\infty$		
	$\varepsilon=0$	$0<\varepsilon<\infty$	$\varepsilon=\infty$	$\varepsilon=0$	$0<\varepsilon<\infty$	$\varepsilon=\infty$	$\varepsilon=0$	$0<\varepsilon<\infty$	$\varepsilon=\infty$
$\alpha=0$	0	0	0	0	0	0	HP	IP	E^{Δ}
$0<\alpha<\infty$	0	0	0	JM	LW	0	E^{∞}	E^{∞}	E^{Δ}
$\alpha=\infty$	0	0	0	E^N	E^N	0 or E^N	E^{∞}	E^{∞}	E^{Δ}

Table 5.1
Boundary cases of the Littlewood model (LW).

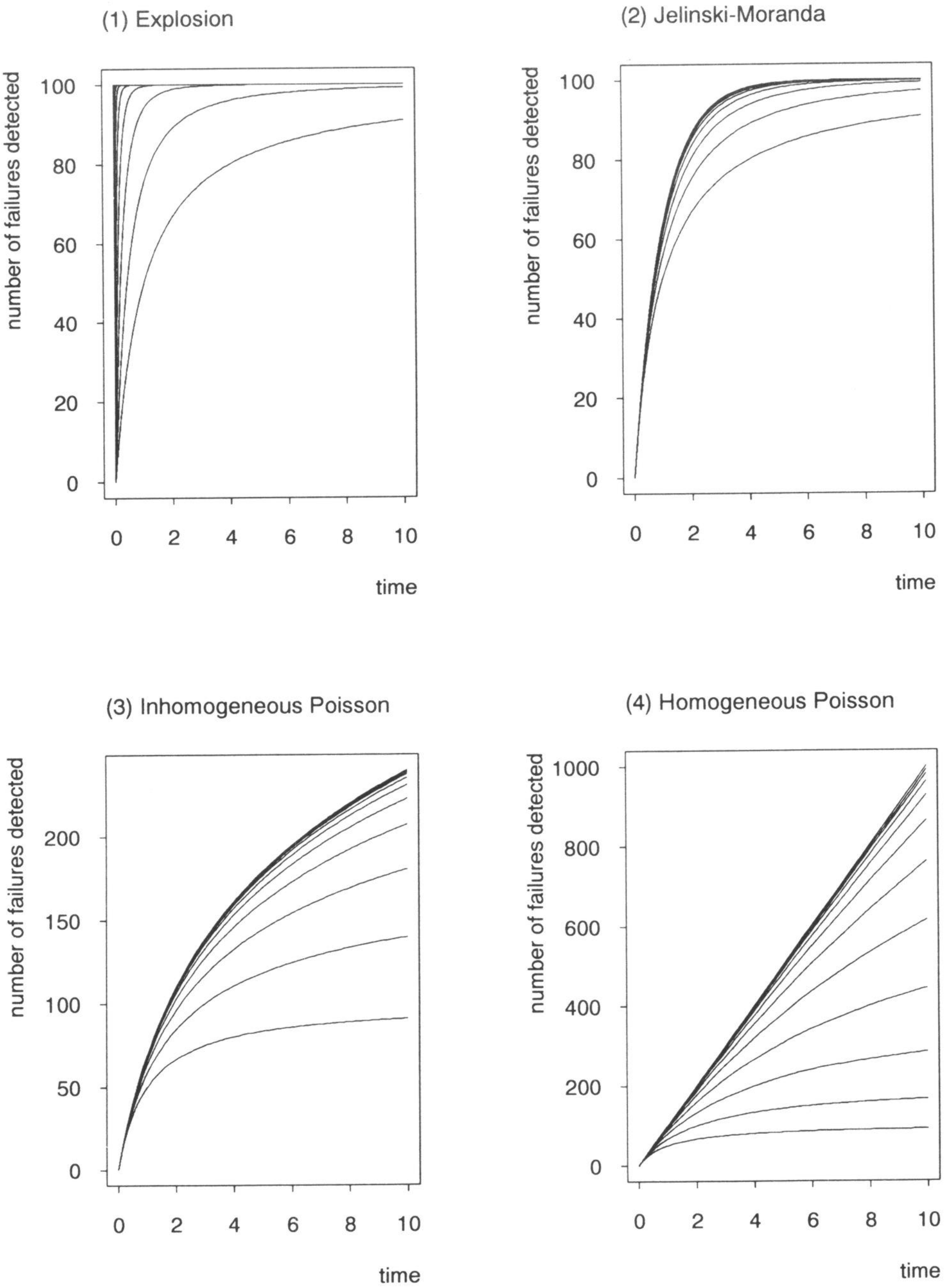


Figure 5.1
Boundary cases of the Littlewood model (LW).

5.3 Maximum likelihood estimation for counting processes

By using the information obtained from the test experiment one can estimate the the parameters of the underlying model. Mostly maximum likelihood estimation is used for this purpose. Aalen (1978) showed using a theorem of Jacod (1975) that the likelihood function for the vector-parameter θ of the stochastic intensity $\lambda^{\mathcal{F}}(t)$ of a counting process $n(t)$ observed on $[0,t]$ and conditioned on the past $\mathcal{F}_{t-}$, is given by

$$L(\theta;t\,|\,\mathcal{F}_{t-}) \;:=\; \exp\left[\int_0^t \log\lambda^{\mathcal{F}}(s;\theta)\,\mathrm{d}n(s) - \int_0^t \lambda^{\mathcal{F}}(s;\theta)\,\mathrm{d}s\right].$$

The past here only consists of the minimal (or self-exciting) history of the counting process $n(t)$:

$$\mathcal{F}_{t-} \;:=\; \sigma\{\, n(s) : 0 \le s < t \,\}.$$

For the Littlewood model with intensity (5.4), parameter $\theta=(N,\alpha,\varepsilon)$ and exposure period $t=\tau$, the log-likelihood function becomes

$$\begin{aligned}\log L(N,\alpha,\varepsilon;\tau) \;:=\; & \sum_{i=1}^{n(\tau)} \log\left[\frac{\alpha(N-i+1)}{1+\varepsilon T_i}\right] - \sum_{i=1}^{n(\tau)+1} \frac{\alpha}{\varepsilon}(N-i+1)\log\left[\frac{1+\varepsilon T_i}{1+\varepsilon T_{i-1}}\right] \\ =\; & n(\tau)\log(\alpha) - \alpha(N-n(\tau))\frac{\log(1+\varepsilon\tau)}{\varepsilon} \\ & + \sum_{i=1}^{n(\tau)} \log(N-i+1) - (\alpha+\varepsilon)\sum_{i=1}^{n(\tau)} \frac{\log(1+\varepsilon T_i)}{\varepsilon}, \end{aligned} \tag{5.5}$$

where $T_0 := 0$ and $T_{n(\tau)+1} := \tau$. Hence the log-likelihood equations are

$$\frac{\partial}{\partial N}\log L(N,\alpha,\varepsilon;\tau) \;=\; \sum_{i=1}^{n(\tau)} \frac{1}{N-i+1} - \alpha\,\frac{\log(1+\varepsilon\tau)}{\varepsilon} = 0, \tag{5.6}$$

$$\frac{\partial}{\partial \alpha}\log L(N,\alpha,\varepsilon;\tau) \;=\; \frac{n(\tau)}{\alpha} - (N-n(\tau))\,\frac{\log(1+\varepsilon\tau)}{\varepsilon} - \sum_{i=1}^{n(\tau)} \frac{\log(1+\varepsilon T_i)}{\varepsilon} = 0, \tag{5.7}$$

$$\begin{aligned}\frac{\partial}{\partial \varepsilon}\log L(N,\alpha,\varepsilon;\tau) \;=\; & \frac{\alpha(N-n(\tau))}{\varepsilon^2}\left[\log(1+\varepsilon\tau) - \frac{\varepsilon\tau}{1+\varepsilon\tau}\right] + \frac{\alpha}{\varepsilon^2}\sum_{i=1}^{n(\tau)}\left[\log(1+\varepsilon T_i) - \frac{\varepsilon T_i}{1+\varepsilon T_i}\right] \\ & - \sum_{i=1}^{n(\tau)} \frac{T_i}{1+\varepsilon T_i} = 0. \end{aligned} \tag{5.8}$$

Remark 5.1 Natural questions to ask are whether $\log L(N,\alpha,\varepsilon;\tau)$, as defined by (5.5), might have more than one (local) maximum, and whether the system of likelihood equations (5.6)-(5.8) might have more than one solution, which is a different problem. The answer to both questions is affirmative. Consider the following data-set:

τ=709.5, $n(\tau)$=3;
$T_1=1, T_2=399.9, T_3=400.1$.

We will see in Remark 5.7 that $\log L(N,\alpha,\varepsilon;\tau)$ has a global maximum at the boundary ($N=n(\tau)$ and $\varepsilon=0$) and both a local maximum and a saddle-point in the interior of the parameter-set Θ. The data-set constructed here is of course in bad agreement with the Littlewood-model. In Remark 5.7 we will consider this data-set in more detail. □

As the system of the three highly non-linear likelihood equations (5.6)-(5.8) obviously cannot be solved analytically Moek (1984) and Geurts *et al.* (1988) use the parametrisation (5.3) and suggest to simplify the problem by fixing one of the parameters $a=1$. Furthermore, Moek (1984) was able to express both N and a as functions of b and used this to derive an equation $f(b)$=0 from which b can be solved numerically relatively easy. Criteria were formulated which ensure the existence of at least one solution. The problem of the possibility of multiple solutions is ignored. Geurts *et al.* (1988) signal this problem but do not provide a solution. They state that in case of multiple solutions exactly one of them will be consistent; moreover this consistent solution is asymptotically normal distributed and efficient. In Sections 5.4-5.7 two approaches are presented to construct estimates that are indeed consistent. Method I in Section 5.4 describes how an initial estimator can be obtained that is $\sqrt{n}$-consistent. Starting with this initial estimator several (iterative) methods will then provide us with estimators that are asymptotically equivalent with the MLE. Another, more analytical approach is presented by Method II in Section 5.6. Here we reduce the problem of maximising the log-likelihood function $\log L(N,\alpha,\varepsilon;\tau)$, given in (5.5), to a one-dimensional one by eliminating first the parameter α explicitly and then the parameter N implicitly. A third approach, based on finding solutions of a slightly modified system of score equations, will be sketched in Section 5.8.

5.4 Method I: Choosing a solution of the likelihood equations

We define $m(t)$ as the expected number of failures detected up to time t. From (5.4) it follows that:

$$m(t) := \mathbb{E}\, n(t) = N\left[1-\left(\frac{1}{1+\varepsilon t}\right)^{\alpha/\varepsilon}\right].$$

In Theorem 5.1 we will prove that given an arbitrary triplet $(s_1,s_2,s_3)\in[0,\tau]^3$ solving

$$m(s_i) = n(s_i)$$

for N, α and ε, will have one solution at most. This solution $\tilde{\theta}$, which can be viewed as a generalised moment estimator, will turn out to be a good starting point for estimating θ_0.

Lemma 5.1 *Consider the function*

$$x(t) = N\left[1-\left(\frac{1}{1+\varepsilon t}\right)^{\alpha/\varepsilon}\right], \quad t\geq 0,\ N>0,\ \alpha>0,\ \varepsilon>0. \tag{5.9}$$

Given three points in time s_1,s_2,s_3, with $0 < s_1 < s_2 < s_3$, the determinant of the Jacobian

$$D := \begin{vmatrix} \frac{\partial}{\partial N}x(s_1) & \frac{\partial}{\partial \alpha}x(s_1) & \frac{\partial}{\partial \varepsilon}x(s_1) \\ \frac{\partial}{\partial N}x(s_2) & \frac{\partial}{\partial \alpha}x(s_2) & \frac{\partial}{\partial \varepsilon}x(s_2) \\ \frac{\partial}{\partial N}x(s_3) & \frac{\partial}{\partial \alpha}x(s_3) & \frac{\partial}{\partial \varepsilon}x(s_3) \end{vmatrix} \tag{5.10}$$

is non-zero.

Proof Lemma 5.1 Fix s_2 and s_3. Consider $D(s_1,s_2,s_3)$ as a function of s_1 only. It is easy to verify that

$$\frac{\partial^2}{\partial s_1^2}D(s_1)$$

has only one zero, hence only 3 zeros for $D(s_1)$. Obviously, however, $D(s_1)$=0 for s_1 equal to 0, s_2 and s_3. As it is given that $0<s_1<s_2<s_3$ it follows immediately, that D is non-zero. □

Theorem 5.1 *We again consider the function $x(t)$ defined by (5.9). Let be given three points in time s_1,s_2,s_3, with $0 < s_1 < s_2 < s_3$ and three positive numbers x_1,x_2,x_3. Then there is at most one parameter-triple (N,α,ε), such that:*

$$x(s_i) = x_i, \quad i=1,2,3. \tag{5.11}$$

Remark 5.2 It is obvious that there are situations in which there exists no solution (N,α,ε) to (5.11). This is the case for instance when monotonicity

$$0 < x_1 < x_2 < x_3 \tag{5.12}$$

or convexity

$$\frac{x_1}{s_1} > \frac{x_2-x_1}{s_2-s_1} > \frac{x_3-x_2}{s_3-s_2} \tag{5.13}$$

are violated. Notice that (5.12) and (5.13) are necessary, but not sufficient conditions for the existence of a solution of (5.11). See also Remark 5.4. □

Remark 5.3 Obviously the result of Theorem 5.1, would follow directly from the non-zeroness of the determinant D, given in (5.10), if we could prove that the boundary of the image of

$$\phi:\ \mathbb{R}^3_+ \rightarrow \mathbb{R}^3:\ (N,\alpha,\varepsilon) \rightarrow (x(s_1),x(s_2),x(s_3))$$

is simply connected. These two facts together, namely, would imply that the mapping ϕ is 1–1 by standard covering arguments (see e.g. Greenberg (1966)). The first result, i.e. the non-zeroness of the determinant, follows from Lemma 5.1 and will be used later on. The connectivity of the boundary, however, is much harder to show. Therefore we have chosen for an alternative, more direct analytical proof, using an appropriate transformation of the parameters N,α and ε. □

Proof Theorem 5.1 Let (N,α,ε) a solution of (4.3). Define

$$u_i := \frac{\log(1+\varepsilon s_i)}{\varepsilon},\ i=1,2,3. \tag{5.14}$$

Then we have

$$x_i = N\left[\ 1-\exp(-\alpha\, u_i)\ \right],\ i=1,2,3;$$

and hence

$$u_i = -\frac{1}{\alpha}\log(1-\frac{x_i}{N}),\ i=1,2,3. \tag{5.15}$$

The symmetry in expressions (5.14) and (5.15) inspires us to define for $i=1,2,3$:

$$v_i(\lambda) := \begin{cases} \dfrac{\log(1+s_i\lambda)}{\lambda} & \text{if } \lambda > 0,\\ s_i & \text{if } \lambda = 0; \end{cases} \tag{5.16}$$

$$w_i(\mu) := \begin{cases} \dfrac{-\log(1-x_i\mu)}{\mu} & \text{if } 0 < \mu < \dfrac{1}{x_3},\\ x_i & \text{if } \mu = 0. \end{cases} \tag{5.17}$$

We now define $V_1:=v_2/v_1$, $V_2:=v_3/v_1$, $W_1:=w_2/w_1$, $W_2:=w_3/w_1$ and consider the two curves in $\mathbb{R}^2$:

$$V(\lambda) := (\ V_1(\lambda)\ ,\ V_2(\lambda)\ ),\quad 0 \le \lambda < \infty, \tag{5.18}$$

$$W(\mu) := (\ W_1(\mu)\ ,\ W_2(\mu)\ ),\quad 0 \le \mu < \frac{1}{x_3}. \tag{5.19}$$

It is easy to check that every intersection of the curves V and W corresponds (in a 1-1 way) with a solution (N, α, ε) of (5.11). We will show now that there exist at most one such crossing, by proving:

Lemma 5.2 *For a crossing, say* (λ_0, μ_0) *of* V *and* W*, given by (5.16)-(5.19), we have*

$$\frac{dV_1}{dV_2}(\lambda_0) < \frac{dW_1}{dW_2}(\mu_0). \tag{5.20}$$

Proof Lemma 5.2 For the proof of Lemma 5.2 we will need two other lemma's.

Lemma 5.3 *The function*

$$f(u) := \frac{(1+u)\log(1+u)}{u}, \quad u > -1$$

is strict monotone increasing.

Proof Lemma 5.3
Calculating the first derivative of f we find

$$f'(u) := \begin{cases} \dfrac{u - \log(1+u)}{u^2} & \text{if } u > -1, \ u \neq 0; \\ \dfrac{1}{2} & \text{if } u = 0. \end{cases}$$

As $u > \log(1+u)$ both for $-1 < u < 0$ and for $u > 0$, we find that $f'(u) > 0$ for all $u > -1$. This proves Lemma 5.3. □

Lemma 5.4 *If* $\lambda_1 < \lambda_2 < \lambda_3$ *and* $v_1 < v_2 < v_3$ *are six real numbers, then*

$$\begin{vmatrix} e^{\lambda_1 v_1} & e^{\lambda_2 v_1} & e^{\lambda_3 v_1} \\ e^{\lambda_1 v_2} & e^{\lambda_2 v_2} & e^{\lambda_3 v_2} \\ e^{\lambda_1 v_3} & e^{\lambda_2 v_3} & e^{\lambda_3 v_3} \end{vmatrix} > 0.$$

Proof Lemma 5.4 Can be found in any textbook on matrix theory, see e.g. Gantmacher (1954). □

We now return to the proof of Lemma 5.2. Differentiating (5.18)-(5.19) yields:

$$\frac{dV_2}{dV_1}(\lambda) := \frac{\dfrac{s_3}{1+s_3\lambda}\log(1+s_1\lambda) - \dfrac{s_1}{1+s_1\lambda}\log(1+s_3\lambda)}{\dfrac{s_2}{1+s_2\lambda}\log(1+s_1\lambda) - \dfrac{s_1}{1+s_1\lambda}\log(1+s_2\lambda)}; \tag{5.21}$$

$$\frac{dW_2}{dW_1}(\mu) := \frac{\frac{-x_3}{1-x_3\mu}\log(1-x_1\mu) - \frac{-x_1}{1-x_1\mu}\log(1-x_3\mu)}{\frac{-x_2}{1-x_2\mu}\log(1-x_1\mu) - \frac{-x_1}{1-x_1\mu}\log(1-x_2\mu)}. \tag{5.22}$$

Now with

$$u_i := \frac{\log(1+s_i\lambda_0)}{\lambda_0}, \quad i=1,2,3.$$

we have

$$\frac{\log(1-x_i\mu_0)}{\log(1-x_1\mu_0)} = \frac{\log(1+s_i\lambda_0)}{\log(1+s_1\lambda_0)} = \frac{u_i}{u_1}, \quad i=2,3.$$

The derivatives (5.21)-(5.22) hence can be written as

$$\frac{dV_2}{dV_1}(\lambda_0) := \frac{\frac{s_3}{1+s_3\lambda_0}u_1 - \frac{s_1}{1+s_1\lambda_0}u_3}{\frac{s_2}{1+s_2\lambda_0}u_1 - \frac{s_1}{1+s_1\lambda_0}u_2}, \tag{5.23}$$

$$\frac{dW_2}{dW_1}(\mu_0) := \frac{\frac{x_3}{1-x_3\mu_0}u_1 - \frac{x_1}{1-x_1\mu_0}u_3}{\frac{x_2}{1-x_2\mu_0}u_1 - \frac{x_1}{1-x_1\mu_0}u_2}. \tag{5.24}$$

From Lemma 5.3 it follows directly that the denominators of (5.23) and (5.24) are respectively negative and positive. So inequality (5.20) is equivalent to

$$\left(\frac{s_3}{1+s_3\lambda_0}u_1 - \frac{s_1}{1+s_1\lambda_0}u_3\right)\left(\frac{x_2}{1-x_2\mu_0}u_1 - \frac{x_1}{1-x_1\mu_0}u_2\right)$$

$$> \left(\frac{s_2}{1+s_2\lambda_0}u_1 - \frac{s_1}{1+s_1\lambda_0}u_2\right)\left(\frac{x_3}{1-x_3\mu_0}u_1 - \frac{x_1}{1-x_1\mu_0}u_3\right) \tag{5.25}$$

and (5.25) is again equivalent to

$$\begin{vmatrix} \dfrac{s_1}{1+s_1\lambda_0} & u_1 & \dfrac{x_1}{1-x_1\mu_0} \\ \dfrac{s_2}{1+s_2\lambda_0} & u_2 & \dfrac{x_2}{1-x_2\mu_0} \\ \dfrac{s_3}{1+s_3\lambda_0} & u_3 & \dfrac{x_3}{1-x_3\mu_0} \end{vmatrix} > 0. \tag{5.26}$$

Notice that for $i=1,2,3$:

$$\frac{s_i}{1+s_i\lambda_0} = \frac{1-\mathrm{e}^{-\lambda_0 u_i}}{\lambda_0} = \int_0^{u_i} \mathrm{e}^{-\lambda_0 v}\, dv, \tag{5.27}$$

$$\frac{x_i}{1-x_i\mu_0} = \frac{\mathrm{e}^{c\lambda_0 u_i}-1}{\mu_0} = \frac{c\lambda_0}{\mu_0}\int_0^{u_i} \mathrm{e}^{c\lambda_0 v}\, dv, \tag{5.28}$$

$$u_i = \int_o^{u_i} 1\, dv. \tag{5.29}$$

Substituting (5.27)-(5.29) in (5.26) yields another equivalent inequality:

$$\frac{c\lambda_0}{\mu_0} \int_{v_1=0}^{u_1} \int_{v_2=u_1}^{u_2} \int_{v_3=u_2}^{u_3} \begin{vmatrix} \mathrm{e}^{-\lambda_0 v_1} & 1 & \mathrm{e}^{c\lambda_0 v_1} \\ \mathrm{e}^{-\lambda_0 v_2} & 1 & \mathrm{e}^{c\lambda_0 v_2} \\ \mathrm{e}^{-\lambda_0 v_3} & 1 & \mathrm{e}^{c\lambda_0 v_3} \end{vmatrix} \mathrm{d}v_3 \mathrm{d}v_2 \mathrm{d}v_1 > 0. \tag{5.30}$$

As a direct consequence of Lemma 5.4 the determinant in the integrand of (5.30) is strictly positive. Therefore inequality (5.30) and equivalently (5.20) are satisfied. Lemma 5.2 is now completely proved. □

We return to the proof of Theorem 5.1. From (5.20) it follows directly that the curves $V(\lambda)$ and $W(\mu)$ can have one intersection at most. This corresponds with at most one parameter-triple (N, α, ε) as a solution to (5.11). Theorem 5.1 is now completely proved. □

Remark 5.4 Define

$$\lambda := \frac{1+\sqrt{5}}{2} \tag{5.31}$$

Set $(s_1,s_2,s_3)=(1,2,3)$ and $(x_1,x_2,x_3)=(2,3,y)$, where

$$3.672276 \sim \frac{2\log(1+3\lambda)}{\log(1+\lambda)} < y < 4. \tag{5.32}$$

Then one can easily verify the monotonicity and convexity conditions (5.12)–(5.13). Despite this, the curves $V(\lambda)$ and $W(\mu)$, defined by (5.16)–(5.19), have no intersection. This example counters the idea that (5.12)–(5.16) would also be sufficient for the existence of such a crossing. We need an extra condition, perhaps embodying the fact that $x'''(t)>0$ for all t. □

It is obvious that maximising $\log L(N,\alpha,\varepsilon;\tau)$, as defined in (5.5), can be very hard, but is not impossible. In Van Pul (1992) we have used a standard optimisation program written in Pascal, called *Amoeba* and described in Vetterling *et al.* (1985), which carries out a fast down-hill simplex method. This method does not treat the problem as a series of one-dimensional maximisations and only function evaluations are involved, not derivatives. As input it requires the four edges of a non-degenerate simplex S in IR^3, a (function) tolerance ε and the function f, which is to be maximised. An appropriate sequence of reflections, expansions and contractions of the initial simplex S *should always converge* to the maximum of the function f, not necessarily lying in the initial simplex S. Guarantees, however, that Amoeba finds the absolute maximum in the presence of local maxima, cannot be given. The existence of more than one (local) maximum, however, is very well possible, although so far multiple solutions have never been found in practice, as we know (see also Moek (1983). The data-set constructed in Remark 3.1 shows, however, that it is theoretically possible. Furthermore it is possible that Amoeba crashes, because the maximum of the function was taken at the boundary of the simplex. In such a case we had to restart the Amoeba procedure with a different initial simplex S'. In the sequel of this paper, we will need an optimisation procedure with nice mathematical properties. Therefore we introduce the following assumption:

Assumption 5.1 There exists an optimisation program, say *Max*, which, when given a continuous, concave function $f:\mathrm{IR}^3 \to \mathrm{IR}$ and a compact subset S of IR^3, *will always return* an $x_{\max} \in S$ with the property that for all $x \in S$:

$$f(x) \le f(x_{\max}).$$

□

The algorithm to find a consistent MLE consists then of the following four steps:

Algorithm 5.1

(step1) Choose three time points $0<s_1<s_2<s_3<\tau$, independent of the the past of the counting process. For instance: $s_i := i\tau/3$. Suppose the number of faults detected up to those time points are n_1, n_2, n_3. We assume all conditions necessary for the existence of a crossing of $V(\lambda)$ and $W(\mu)$ are satisfied.

(step2) One can determine numerically estimators $\tilde{\lambda}$ for λ_0 and $\tilde{\mu}$ for μ_0, such that $V(\tilde{\lambda}) = W(\tilde{\mu})$.

(step3) Compute $\tilde{N}$, $\tilde{\alpha}$ and $\tilde{\varepsilon}$ by

$$\tilde{N} := \frac{1}{\tilde{\mu}},$$

$$\tilde{\alpha} := -\tilde{\lambda}\frac{\log(1-x_i\tilde{\mu})}{\log(1+s_i\tilde{\lambda})} \qquad (5.33)$$

and

$$\tilde{\varepsilon} := \tilde{\lambda}.$$

Notice that $\tilde{\alpha}$ in (5.33) is independent of i. Define the initial estimator $\tilde{\theta} := (\tilde{N}, \tilde{\alpha}, \tilde{\varepsilon})$.

(step4) Construct a regular tetrahedron around $\tilde{\theta}$ with vertex length equal to $Cn(\tau)^{-1/3}$ and apply the optimisation program *Max* (hypothesised in Assumption 5.1) to the (locally concave!) log-likelihood function (5.5) on this tetrahedron. With a probability tending to one, the consistent maximum likelihood estimator $\hat{\theta}$ will be in this tetrahedron too and if so *Max* will find it. □

Remark 5.5 If such a crossing as mentioned in step 1 does not exist, we could repeat this step with new s_i, but the theory developed could then not be applied; for the theory it is namely essential that the choice of the s_i is independent of the data. Note that if it is very hard to find a triple (s_1, s_2, s_3), satisfying the necessary conditions, then one might suspect that the data reject the Littlewood model decisively. It is then, of course, of no use searching for accurate estimators. □

5.5 Asymptotic properties of Method I estimator

In order to discuss the statistical properties of the estimator, we have to give some more background. [Here follows a summary of the concepts and results of Chapter 3.] Important will be the way in which we will treat asymptotics. It does not make sense to let τ, the stopping time, grow to infinity. In the long run the estimate of the total number of faults will trivially be equal to the true number of faults. It makes more sense to (conceptually) increase the number of faults in the program. The idea is that then asymptotics should be relevant to the practical situation in which N_0 is large and $n(\tau)/N_0$ not close to zero or one.

Let a counting process $n(t)$ be given. The jumps of the counting process $n(t)$ are observed only during a specific time interval $[0,\tau]$. In this paper we will assume that the intensity function associated with the counting process exists and is a member of some specified parametric family, that is:

$$\lambda(t) := \lambda(t; N, \psi), \quad t \in [0,\tau],\ N \in \mathrm{IN},\ \psi \in \Psi,\ \Psi \subset \mathrm{IR}^{p-1}$$

for an integer p. Let N_0 and ψ_0 be the true parameter values. Typically the parameter N_0 represents the scale or size of the problem (sometimes $N_0 = n(\infty)$), while ψ_0 is a nuisance vector parameter. We will be interested in estimation of N_0 and ψ_0 as $N_0 \to \infty$. We assume that the model is also meaningful for non-integer N. For instance the intensity function (2.2) of the Jelinski-Moranda model can be generalised to

$$\lambda^{JM}(t) = \phi\left(N - n(t-)\right) I\{ n(t) < N \}, \quad t \in [0,\tau],$$

where $I\{.\}$ denotes the indicator function. As we are particularly interested in the parameter estimation when N_0 is large, we will introduce a series of counting processes $n_\nu(t)$, $t \in [0,\tau], \nu = 1,2,\ldots$ and let N conceptually increase. Let $N = N_\nu \to \infty$ for $\nu \to \infty$. By the reparametrisation

$$N_\nu = \nu \gamma_\nu,$$

with a dummy variable γ_ν, we can denote the associated intensity functions by

$$\lambda_\nu(t;\gamma,\psi) := \lambda(t;\nu\gamma,\psi), \quad t \in [0,\tau], \quad \gamma \in \mathrm{IR}^+, \quad \psi \in \Psi, \quad \nu = 1,2,\ldots.$$

Now we consider the estimation of γ and ψ as $\nu \to \infty$. If the real-life situation has $\nu = N_0$, then $\gamma = \gamma_0 = 1$ and $\psi = \psi_0$. It is rather unorthodox to increase a model parameter itself, in this case N. This complication is solved by estimating γ. We will assume that the maximum likelihood estimators $(\hat{\gamma}_\nu, \hat{\psi}_\nu)$ for (γ_0, ψ_0) exist. Typically, $(\hat{\gamma}_\nu, \hat{\psi}_\nu)$ is a root of the likelihood equations

$$\frac{\partial}{\partial(\gamma,\psi)} \log L_\nu(\gamma,\psi;\tau) = 0, \quad \nu = 1,2,\ldots, \tag{5.34}$$

where the likelihood function at time t $L_\nu(\gamma,\psi;t)$ is given by (see Aalen (1978))

$$L_\nu(\gamma,\psi;t) := \exp\left(\int_0^t \log \lambda_\nu(s;\gamma,\psi)\, dn_\nu(s) - \int_0^t \lambda_\nu(s;\gamma,\psi)\, ds \right). \tag{5.35}$$

We define for $\nu = 1,2,\ldots$ the stochastic process $x_\nu(t)$ by

$$x_\nu(t) := \nu^{-1} n_\nu(t), \quad t \in [0,\tau].$$

In most practical situations, this sequence of stochastic processes converges uniformly on $[0,\tau]$ in probability to a deterministic function $x_0(t)$ as $\nu \to \infty$ (Kurtz (1983)). We assume that the counting processes n_ν are generated by associated intensity functions $\lambda_\nu(t)$, satisfying

$$\lambda_\nu(t) = \nu \beta(t;\theta; x_\nu(t-)),$$

for an arbitrary non-negative and non-anticipating function $\beta:[0,\tau]\times\Theta\times K \rightarrow \mathrm{IR}^+$ where the model parameter $\theta=(\gamma,\psi)$ consists of the parameter of most interest γ and a nuisance parameter vector ψ. Under classical smoothness and boundedness conditions on the function β (see for instance Borgan (1984) or Van Pul (1992)), we have the following result:

Theorem 5.2 *Consistency of ML-estimators: With a probability tending to 1, the likelihood equations (5.34) have exactly one consistent solution $\hat{\theta}_\nu$. Moreover, this solution provides a local maximum of the likelihood function (5.35), has an asymptotically normal distribution and is efficient.*

This theorem and its proof can be found in Van Pul (1992). We are now able to formulate the following results:

Theorem 5.3 $\tilde{\theta}$ *is* $\sqrt{n}$*-consistent.*

Proof Theorem 5.3 We first prove consistency of $\tilde{\theta}$. From Lemma 5.1 and Theorem 5.1 it follows directly, that we can write

$$\tilde{\theta} = g(x),$$

for some function g: $\mathrm{IR}^3 \rightarrow \Theta$, with continuous partial derivatives. Let $\tilde{\theta}_0 := g(x_0)$ and $\tilde{\theta}_\nu := g(x_\nu)$, where we use the asymptotics developed in the beginning of this section. Then by the first theorem of Kurtz (1981) we have

$$x_\nu \rightarrow x$$

in probability uniformly in t on $[0,\tau]$. Hence by the continuity of g we deduce the consistency:

$$\tilde{\theta}_\nu - \tilde{\theta}_0 = g(x_\nu) - g(x_0) \rightarrow_{\mathrm{IP}} 0,$$

as $\nu \rightarrow \infty$. The $\sqrt{n}$-consistency now follows immediately from Kurtz's second theorem (1981) and an application of the delta method. This proves Theorem 5.3. □

Corollary 5.1 *Algorithm 5.1 works, that is, it yields an efficient estimator for* θ_0.

Corollary 5.2 *The maximum likelihood estimator* $\hat{\theta}$ *with the smallest Euclidean distance to* $\tilde{\theta}$ *is consistent.*

Corollary 5.3 *The k-step Newton-Raphson modification of* $\tilde{\theta}$ *is asymptotically equivalent to the MLE* $\hat{\theta}$.

As we do not know of the existence of an optimisation algorithm, like *Max,* satisfying Assumption 5.1, to apply Corollary 5.1 we make use of the program *Amoeba* instead. In practice, this yields nice results (see Van Pul (1992)). Corollary 5.2 is a purely theoretical result. In general there are no numerical recipes available that deliver all solutions of a system of non-linear equations. Corollary 5.3 is frequently used in practical situations, but results often turn out to be very disappointing, especially for $k=1$. For $k=4$ or $k=5$ this method should however work (see Andersen *et al.* (1993)).

5.6 Method II: Exploiting the mathematical properties of the likelihood function

For sake of convenience we apply the parameter-transformation

$$M := N - n(\tau).$$

The log-likelihood, as expressed in the new parameters, is given by

$$\log L(\alpha,M,\varepsilon) = n(\tau)\log\alpha - (\varepsilon+\alpha)\sum_{i=1}^{n(\tau)} \frac{\log(1+\varepsilon T_i)}{\varepsilon} - \alpha M \frac{\log(1+\varepsilon\tau)}{\varepsilon} + \sum_{i=1}^{n(\tau)} \log(M+i) \quad (5.39)$$

with parameter-domain $\{ (\alpha, M, \varepsilon) : \alpha>0, M\geq 0, \varepsilon\geq 0 \}$. When M and ε are kept fixed, it is quite easy to maximise $\log L(\alpha,M,\varepsilon)$; it can be done analytically. If we define the profile-likelihood

$$R_2(M,\varepsilon) := \sup_{\alpha>0} \log L(\alpha,M,\varepsilon);$$

then

$$\sup_{\alpha>0, M\geq 0, \varepsilon\geq 0} \log L(\alpha,M,\varepsilon) = \sup_{M\geq 0,\varepsilon\geq 0} R_2(M,\varepsilon).$$

Now

$$\frac{\partial \log L}{\partial\alpha} = \frac{n(\tau)}{\alpha} - \sum_{i=1}^{n(\tau)} \frac{\log(1+\varepsilon T_i)}{\varepsilon} - M\frac{\log(1+\varepsilon\tau)}{\varepsilon}$$

and it is easy to check that $\log L(\alpha,M,\varepsilon)$ for M, ε fixed, is maximal for

$$\alpha = n(\tau)\left[\sum_{i=1}^{n(\tau)} \frac{\log(1+\varepsilon T_i)}{\varepsilon} + M\frac{\log(1+\varepsilon\tau)}{\varepsilon}\right]^{-1}; \quad (5.40)$$

so

$$R_2(M,\varepsilon) = n(\tau)\log n(\tau) - n(\tau)\log\left[\sum_{i=1}^{n(\tau)} \frac{\log(1+\varepsilon T_i)}{\varepsilon} + M\frac{\log(1+\varepsilon\tau)}{\varepsilon}\right]$$

$$- \sum_{i=1}^{n(\tau)} \log(1+\varepsilon T_i) + \sum_{i=1}^{n(\tau)} \log(M+i). \quad (5.41)$$

In order to maximise $R_2(M, \varepsilon)$ we will first maximise with respect to M, keeping ε fixed, and then maximise with respect to ε. The first maximisation is relatively easy because we can make use of the following theorem:

Theorem 5.4 *For n integer and ≥ 2, and η real and >0, the function*

$$h(M) := -n\log(M+\eta) + \sum_{i=1}^{n} \log(M+i)$$

has precisely one local extremum, which is realised at:

- $M=0$, *for* $\eta \leq n[\sum_{i=1}^{n} \frac{1}{i}]^{-1}$,
- $M=\infty$, *for* $\eta \geq \frac{n+1}{2}$,
- $M \in (0,\infty)$, *for intermediate values of* η.

Proof Theorem 5.4 It is easy to check that

$$\begin{aligned} \frac{d}{dM} h(M) &= \frac{-n}{M+\eta} + \sum_{i=1}^{n} \frac{1}{M+i}, \\ \frac{d^2}{dM^2} h(M) &= \frac{n}{(M+\eta)^2} - \sum_{i=1}^{n} \left(\frac{1}{M+i} \right)^2 . \end{aligned}$$

If for a finite value M' of M

$$\frac{d}{dM} h(M') = 0, \tag{5.42}$$

then

$$\begin{aligned} \frac{d^2}{dM^2} h(M') &= \frac{n}{(M'+\eta)^2} - \sum_{i=1}^{n} \left(\frac{1}{M'+i} \right)^2 \\ &= \frac{1}{n} \left(\sum_{i=1}^{n} \frac{1}{M'+i} \right)^2 - \sum_{i=1}^{n} \left(\frac{1}{M'+i} \right)^2 \\ &< 0, \end{aligned} \tag{5.43}$$

since $n \geq 2$. So there is at most one finite value M' for M for which (5.42) holds, because if there were two of them, say $M'_1 < M'_2$, then we would have $\frac{d^2}{dM^2} h(M) \geq 0$ for either one of them, or there would be an intermediate value M'_3, $M'_1 < M'_3 < M'_2$ for which $\frac{d}{dM} h(M'_3) = 0$, $\frac{d^2}{dM^2} h(M'_3) \geq 0$. Both options are, however, excluded by (5.43).

- Case 1: $\eta \leq n \left[\sum_{i=1}^{n} \frac{1}{i} \right]^{-1}$.

  It is easy to check that $\frac{\mathrm{d}}{\mathrm{d}M} h(0) \leq 0$. From this and (5.43) it follows immediately that $\frac{\mathrm{d}}{\mathrm{d}M} h(M) < 0$ for all $M > 0$. Hence $M = 0$ provides the unique local maximum of $h(M)$.

- Case 2: $\eta \geq \frac{n+1}{2}$.

  We find that for all $M \geq 0$:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}M} h(M) &= \frac{-n}{M+\eta} + \sum_{i=1}^{n} \frac{1}{M+i} \\ &\geq \frac{-n}{M + \frac{n+1}{2}} + \sum_{i=1}^{n} \frac{1}{M+i} \\ &\geq 0. \end{aligned}$$

  So the supremum is reached at $M = \infty$.

- Case 3: $n \left[\sum_{i=1}^{n} \frac{1}{i} \right]^{-1} < \eta < \frac{n+1}{2}$.

  We have $\frac{\mathrm{d}}{\mathrm{d}M} h(0) > 0$, but

$$\lim_{M \to \infty} M^2 \frac{\mathrm{d}}{\mathrm{d}M} h(M) = n\eta - \sum_{i=1}^{n} i < 0.$$

  So there exists a M_0, $0 < M_0 < \infty$, such that for all $M > M_0$ we have $M^2 \frac{\mathrm{d}}{\mathrm{d}M} h(M) < 0$ and hence also $\frac{\mathrm{d}}{\mathrm{d}M} h(M) < 0$. Thus again there exists a finite positive value M' for which $\frac{\mathrm{d}}{\mathrm{d}M} h(M') = 0$ and the unique local maximum of the function $h(M)$ is realised.

This completes the proof of Theorem 5.4. □

Now $R_2(M, \varepsilon)$ can be rewritten as

$$\begin{aligned} R_2(M,\varepsilon) &= n(\tau)\log n(\tau) - n(\tau)\log\left(\frac{\log(1+\varepsilon\tau)}{\varepsilon}\right) - n(\tau)\log(M+\eta(\varepsilon)) \\ &\quad + \sum_{i=1}^{n(\tau)} \log(M+i) - \sum_{i=1}^{n(\tau)} \log(1+\varepsilon T_i), \end{aligned} \tag{5.44}$$

where

$$\eta(\varepsilon) := \sum_{i=1}^{n(\tau)} \frac{\log(1+\varepsilon T_i)}{\log(1+\varepsilon\tau)}.$$

Only the third and the fourth of the terms in the right-hand side of (5.44) depend on M. Their sum has a form similar to the function $h(M)$ of Theorem 5.4, and so $R_2(M,\varepsilon)$ has for ε fixed, exactly one local supremum, which occurs for $M=0$ if $\eta(\varepsilon)\le n[\Sigma 1/i]^{-1}$, for $M=\infty$ if $\eta(\varepsilon)\ge(n+1)/2$ and for a finite, positive value of M otherwise.

Remark 5.6 $\tilde{M}(\varepsilon)$, the value of M which maximises $R_2(M,\varepsilon)$, cannot be expressed explicitly, contrary to the α of (5.40). If ε increases, $\eta(\varepsilon)$ increases as well, and so will $\tilde{M}(\varepsilon)$. □

Finally then, if we define

$$R_1(\varepsilon) := R_2(\tilde{M}(\varepsilon),\varepsilon),$$

we have to maximise $R_1(\varepsilon)$. While doing this, one must bear in mind that $R_1(\varepsilon)$ might reach its maximum for $\varepsilon=0$. As a matter of fact, we have encountered this feature several times when analyzing real data-sets.

Remark 5.7 We now return to the data-set mentioned in Remark 5.1:

$n(\tau)=3;$
$T_1=1, T_2=399.9, T_3=400.1;$
$\tau=709.5.$

The function $R_1(\varepsilon)$ takes its global maximum for $\varepsilon=0$, a second local maximum for $\varepsilon=1.152$ and a local minimum for $\varepsilon=0.023$. This corresponds to a global maximum for $R_2(M,\varepsilon)$ for $\varepsilon=0$, $M=0$, a second local maximum for $R_2(M,\varepsilon)$ for $\varepsilon=1.152$, $M=9.8$, and a saddle-point at $\varepsilon=0.023$, $M=0.0015$. □

All maximisation results are easily obtained by a standard optimisation procedure for one-dimensional functions, called *Golden section search* (see for instance Vetterling *et al.* (1985)).

5.7 Asymptotic properties of Method II estimator

We have the following important result:

Theorem 5.5 *The global maximiser of the log-likelihood function for the Littlewood model (see expressions (5.5) or (5.39)) is consistent.*

Proof Theorem 5.5 It will turn out to be of great advantage to apply the parameter-transformation

$$\zeta := \frac{\alpha \log(1+\varepsilon\tau)}{\varepsilon}. \tag{5.45}$$

With the new parameters N, ε and ζ, we can rewrite the log-likelihood as follows:

$$\log L(N,\varepsilon,\zeta) = n(\tau)\log\frac{\zeta}{1-e^{-\zeta}} - \zeta\sum_{i=1}^{n(\tau)}\frac{\log(1+\varepsilon t_i)}{\log(1+\varepsilon\tau)}$$

$$+ n(\tau)\log\frac{\varepsilon}{\log(1+\varepsilon\tau)} - \sum_{i=1}^{n(\tau)}\log(1+\varepsilon t_i)$$

$$+ n(\tau)\log(1-e^{-\zeta}) - (N-n(\tau))\zeta + \sum_{i=1}^{n(\tau)}\log(N-i+1)$$

$$= \sum_{i=1}^{n(\tau)}\log h(t_i;\varepsilon,\zeta)$$

$$+ n(\tau)\log(1-e^{-\zeta}) - (N-n(\tau))\zeta + \sum_{i=1}^{n(\tau)}\log(N-i+1), \tag{5.46}$$

where

$$h(t;\varepsilon,\zeta) := \frac{\zeta}{1-e^{-\zeta}}\exp\left(-\zeta\frac{\log(1+\varepsilon t)}{\log(1+\varepsilon\tau)}\right)\frac{\varepsilon}{\log(1+\varepsilon\tau)}\frac{1}{1+\varepsilon t}.$$

It is easy to check that $h(t;\varepsilon,\zeta)$ is continuous on $[0,\tau]$ and for all $\varepsilon\geq 0,\zeta\geq 0$:

$$\int_0^\tau h(s;\varepsilon,\zeta)\mathrm{d}s = 1.$$

Hence $h(t;\varepsilon,\zeta)$ can be considered as the continuous density function of a stochastic variable with values in $[0,\tau]$. We are now going to eliminate N from the log-likelihood by maximising (5.46) with respect to N for fixed ε and ζ. Let $\hat{N}(\zeta)$ be that value of $N\geq n(\tau)$, for which

$$R(N,\zeta) := n(\tau)\log(1-e^{-\zeta}) - (N-n(\tau))\zeta + \sum_{i=1}^{n(\tau)}\log(N-i+1)$$

is maximal and define

$$R(\zeta) \ := \ n(\tau)\log(1-\mathrm{e}^{-\zeta}) \ - \ (\hat{N}(\zeta) - n(\tau))\zeta \ + \ \sum_{i=1}^{n(\tau)} \log(\hat{N}(\zeta)-i+1).$$

For large values of $n(\tau)$, $R(\zeta)$ is approximately $n(\tau)\log n(\tau)-n(\tau)+\zeta/2$, but we will not need this. We will only use that:

$$0 \le \frac{\partial}{\partial\zeta}R(\zeta) \le 1. \tag{5.47}$$

Now the log-profile-likelihood for (ε,ζ) becomes

$$LPL(\varepsilon,\zeta) \ := \ \sum_{i=1}^{n(\tau)} \log h(t_i;\varepsilon,\zeta) \ + \ R(\zeta).$$

As in our asymptotic approach $N_0 \to \infty$ implies $n(\tau) \to \infty$ with probability 1, we consider the sequence $(T_i)_{i\ge1}$ of i.i.d random variables with density $h(t,\varepsilon_0,\zeta_0)$. We will prove that the corresponding sequence of profile-log-likelihood maximisers $(\hat{\varepsilon}_n,\hat{\zeta}_n)$ converges to (ε_0,ζ_0) with probability 1. With $\Omega = \{\ \varepsilon,\zeta) : \varepsilon\ge0,\ \zeta\ge0\ \}$, the density function $h(t;\varepsilon,\zeta)$ satisfies all eight assumptions of Wald (1949). The verification of the validity of these assumptions is straightforward, except perhaps for Assumptions II and V. Now it follows from a theorem of Wald (1949) that for all closed subsets ω of Ω, not containing the true value (ε_0,ζ_0), we have

$$\mathrm{IP}\left[\lim_{n\to\infty} \sup_{(\varepsilon,\zeta)\in\omega} \left[\sum_{i=1}^{n} \log h(t_i;\varepsilon,\zeta) \ - \ \sum_{i=1}^{n} \log h(t_i;\varepsilon_0,\zeta_0)\right] \ = \ -\infty\right] \ = \ 1.$$

Some calculations yield that with probability 1 there exists a N_1 such that for $n > N_1$:

$$\hat{\zeta}_n \ < \ \frac{4\tau}{\mathrm{IE}T}, \tag{5.48}$$

where $\mathrm{IE}T$ denotes the expectation of the stochastic variable with density $h(t;\varepsilon_0,\zeta_0)$. As $LPL(\hat{\varepsilon}_n,\hat{\zeta}_n) \ge LPL(\varepsilon_0,\zeta_0)$ it follows that for $n > N_1$:

$$\begin{aligned}\sum_{i=1}^{n} \log h(t_i;\hat{\varepsilon}_n,\hat{\zeta}_n) \ - \ \sum_{i=1}^{n} \log h(t_i;\varepsilon_0,\zeta_0) \ &\ge \ R(\zeta_0) \ - \ R(\hat{\zeta}_n) \\ &\ge \ \min(0,\zeta_0 - \hat{\zeta}_n) \\ &\ge \ -\frac{4\tau}{\mathrm{IE}T}.\end{aligned}$$

Let $\eta>0$ be arbitrary. Suppose the sequence $(\hat{\varepsilon}_n,\hat{\zeta}_n)$ has an accumulation point at a distance larger than η of (ε_0,ζ_0). So for infinitely many n:

$$\sup_{|(\varepsilon,\zeta)-(\varepsilon_0,\zeta_0)|\,>\,\eta} \sum_{i=1}^{n} \log h(t_i;\varepsilon,\zeta) \;\geq\; \sum_{i=1}^{n} \log h(t_i;\hat{\varepsilon}_n,\hat{\zeta}_n)$$

and hence

$$\sup_{|(\varepsilon,\zeta)-(\varepsilon_0,\zeta_0)|\,>\,\eta} \sum_{i=1}^{n} \log h(t_i;\varepsilon,\zeta) \;\geq\; \sum_{i=1}^{n} \log h(t_i;\varepsilon_0,\zeta_0) - \frac{4\tau}{\mathrm{I\!E}T}$$

for infinitely many n. But this event has according to Wald's theorem probability zero. Therefore all accumulation points of $(\hat{\varepsilon}_n,\hat{\zeta}_n)$ are within distance η of (ε_0,ζ_0). As η was chosen arbitrarily, it follows that

$$\mathrm{I\!P}\left[\lim_{n\to\infty} (\hat{\varepsilon}_n,\hat{\zeta}_n) = (\varepsilon_0,\zeta_0)\right] = 1.$$

This yields the consistency of $(\hat{\varepsilon}_n,\hat{\zeta}_n)$. Of course $\hat{N}(\hat{\zeta}_n)$ itself is not consistent. The best we can hope for and get, as

$$0 < \frac{n(\tau)}{1-e^{-\zeta}} - \hat{N}(\zeta) < 1,$$

is

$$\frac{\hat{N}(\hat{\zeta}_n)}{N_0} \rightarrow_{a.s} \frac{\mathrm{I\!E}(n(\tau))}{N_0}\,\frac{1}{1-e^{-\zeta}} = 1.$$

Full details of the verification of (5.47), (5.48) and Assumptions II and V of Wald can be found in Barendregt & Van Pul (1993). This completes the proof of Theorem 5.5. □

5.8 Method III: Investigating solutions of modified score equations

Recently, the proof of Theorem 5.5 inspired us to a third alternative way of constructing estimators for the model parameters of the Littlewood model. We believe this method, which exhales Le Cam's idea to investigate slightly modified but more easily analysable score equations, will be more generally applicable.

Let us consider again the counting process $n(t)$ with failure intensity according to the Littlewood model:

$$\lambda(t) := \frac{\alpha_0(N_0 - n(t-))}{1+\varepsilon_0 t}.$$

The log-likelihood function in terms of N,α and ε is given by (see also (5.5))

$$\log L\,(N,\alpha,\varepsilon) \;=\; \sum_{i=1}^{n(\tau)} \log\left[\alpha(N-i+1)\right] - (1+\frac{\alpha}{\varepsilon})\sum_{i-1}^{n(\tau)} \log(1+\varepsilon t_i) - \frac{\alpha(N-n\,(\tau))}{\varepsilon}\log(1+\varepsilon\tau). \quad (5.49)$$

In this section we will construct approximate maximum likelihood estimators, sometimes called *modified score equation solutions,* $(\overline{N},\overline{\alpha},\overline{\varepsilon})$ and prove the consistency of $(\overline{\alpha},\overline{\varepsilon})$ and the convergence (in probability) to zero of $\mathrm{var}(\overline{N})/\overline{N}^2$, when N_0 (the true value of N) tends to infinity. We therefore rewrite the log-likelihood as follows:

$$\log L\,(N,\alpha,\varepsilon) \;=\; \sum_{i=1}^{n(\tau)} \log g\,(t_i;\alpha,\varepsilon)$$

$$+n(\tau)\log\left[\frac{1-(1+\varepsilon\tau)^{-\alpha/\varepsilon}}{\alpha}\right] + \sum_{i=1}^{n(\tau)} \log\left[\alpha(N-i+1)\right] - \frac{\alpha(N-n\,(\tau))}{\varepsilon}\log(1+\varepsilon\tau), \quad (5.50)$$

where

$$g\,(t\,;\alpha,\varepsilon) \;:=\; \frac{\alpha(1+\varepsilon t)^{-\alpha/\varepsilon-1}}{1-(1+\varepsilon\tau)^{-\alpha/\varepsilon}}, \quad 0 \le t \le \tau. \quad (5.51)$$

Thus $g\,(t\,;\alpha,\varepsilon)$ is the density function of a continuous stochastic variable with values in $[0,\tau]$. This is true for all parameter combinations (α,ε) with $\alpha \ge 0$ and $\varepsilon \ge 0$. Degeneration of the probability distribution only occurs if at least one of the parameters α and ε takes the value ∞. For $\alpha > 0$, $g\,(t\,;\alpha,\varepsilon)$ can be viewed as the density function of a generalised Pareto distributed variable, that is truncated at τ. We estimate (α,ε) by maximising the leading term of the log-likelihood (5.50):

$$\sum_{i=1}^{n(\tau)} \log g\,(t_i;\alpha,\varepsilon). \quad (5.52)$$

The approximate estimator $(\overline{\alpha},\overline{\varepsilon})$ we obtain in this way is asymptotically consistent, even if we take for the parameter-set $\Omega = \{\,(\alpha,\varepsilon){:}\alpha{\ge}0,\varepsilon{\ge}0\,\}$. This follows directly as the truncated Pareto density $g\,(t\,;\alpha,\varepsilon)$ in (5.51) satisfies all eight assumptions of Wald (1949). Verification of this is, apart from conditions (II) and (V) perhaps, straightforward. Estimation of N could be performed by substituting $\overline{\alpha}$ and $\overline{\varepsilon}$ in the remaining terms of the log-likelihood (5.50):

$$n(\tau)\log\left[\frac{1-(1+\varepsilon\tau)^{-\alpha/\varepsilon}}{\alpha}\right] + \sum_{i=1}^{n(\tau)} \log\left[\alpha(N-i+1)\right] - \frac{\alpha(N-n\,(\tau))}{\varepsilon}\log(1+\varepsilon\tau) \quad (5.53)$$

and maximisation of (5.53) with respect to N. This would yield

$$\overline{N} \;:=\; \frac{n\,(\tau)}{1-(1+\overline{\varepsilon}\tau)^{-\overline{\alpha}/\overline{\varepsilon}}} - \frac{1}{2} + O\,(\frac{1}{n\,(\tau)}). \quad (5.54)$$

We will estimate N by neglecting the last two terms in the right hand side of 5.54. Stated more generally, the idea is to regard the failure times T_i, conditional on $n(\tau)$, as the ordered outcomes of $n(\tau)$ independent variables from some parametric distribution function $F(t,\psi)$. For the Littlewood model this distribution function is the generalised *Pareto* (t,α,ε) distribution. Conditioned on $\psi_0=(\alpha_0,\varepsilon_0)$, $n(\tau)$ has a binomial distribution with parameters N_0 and $F(\tau,\psi_0)$. We define $\bar{\psi}$ and $\bar{N}$ as the roots of

$$\frac{\partial}{\partial\psi}\sum_{i=1}^{n(\tau)}\log\frac{f(T_i,\psi)}{F(\tau,\psi)} = 0, \tag{5.55}$$

$$n(\tau) - N\,F(\tau,\bar{\psi}) = 0. \tag{5.56}$$

Although this choice seems rather *ad hoc,* a closer examination yields that the system of equation (5.55)-(5.56) is in some sense asymptotically equivalent to the system of likelihood equations. Let $\hat{U}_\nu$ be the system of likelihood equations (5.34) and let $\bar{U}_\nu$ be the vector of left-hand sides of (5.55)-(5.56). So $\hat{U}_\nu(\hat{\theta}_\nu)=0$ and $\bar{U}_\nu(\bar{\theta}_\nu)=0$ and furthermore we have

$$\bar{U}_\nu(\theta) = A_\nu(\theta)\hat{U}_\nu(\theta) + B_\nu(\theta),$$

where $A_\nu \to A$, A non-singular and $\sqrt{n}B_n \to 0$. Under weak boundedness and smoothness conditions on A and B the asymptotic properties of $\hat{\theta}$ are transferred to $\bar{\theta}$. More details and numerical results of this method III can be found in Barendregt & Van Pul (1993).

5.9 Comparison of some numerical results of methods I, II and III

In this section we will discuss the results of some simulation experiments. We generated failure times according to the Littlewood model (5.4) with $\alpha_0=1$, $\varepsilon_0=1$ and different values for N_0 (100, 1000 and 10000). The exposure period was kept fixed: $\tau=4$. We compare the initial estimator $\tilde{\theta}$ (suggested in step 3 on page 101), a 5-step Newton-Raphson modification θ^{nr} applied to $\tilde{\theta}$, the global maximiser of the likelihood $\hat{\theta}$ and the modified score equations solution $\bar{\theta}$. Note that the initial estimator $\tilde{\theta}$ does not exist for all data-sets (see Remarks 5.2 and 5.4). For each value of N_0 we repeated the simulation experiments and parameter estimations $K_0=1000$ times. In Table 5.2, K denotes the actual number of cases in which the curves $V(\lambda)$ and $W(\mu)$ intersected each other, leading to an initial estimator $\tilde{\theta}$.

N_0	K
100	521
1000	940
10000	1000

Table 5.2
Number of successful trials to construct $\tilde{\theta}$.

For each value N_0 we picked 200 data-sets randomly from those leading to an initial estimator $\tilde{\theta}$, and computed the associated θ^{nr}, $\hat{\theta}$ and $\bar{\theta}$. Mean square errors for these four estimators are given in Table 5.3.

	(a) N_0=100	(b) N_0=1000	(c) N_0=10000
$\tilde{N}$	1291	32.95	0.0048
$\tilde{\alpha}$	0.3540	0.0205	0.0011
$\tilde{\varepsilon}$	37.04	2.409	0.1206
N^{nr}	23.36	2.599	0.0083
α^{nr}	0.2528	0.0384	0.0017
ε^{nr}	34.91	2.487	0.1206
$\hat{N}$	0.0237	0.0060	0.0006
$\hat{\alpha}$	0.0461	0.0052	0.0007
$\hat{\varepsilon}$	0.5767	0.1322	0.0101
$\bar{N}$	0.1601	0.0099	0.0041
$\bar{\alpha}$	0.0644	0.0075	0.0017
$\bar{\varepsilon}$	0.5473	0.1517	0.0545

Table 5.3
Mean square errors for $\tilde{\theta}$, θ^{nr}, $\hat{\theta}$ and $\bar{\theta}$ with
(a) θ_0=(100, 1, 1), (b) θ_0=(1000, 1, 1), (c) θ_0=(10000, 1, 1).

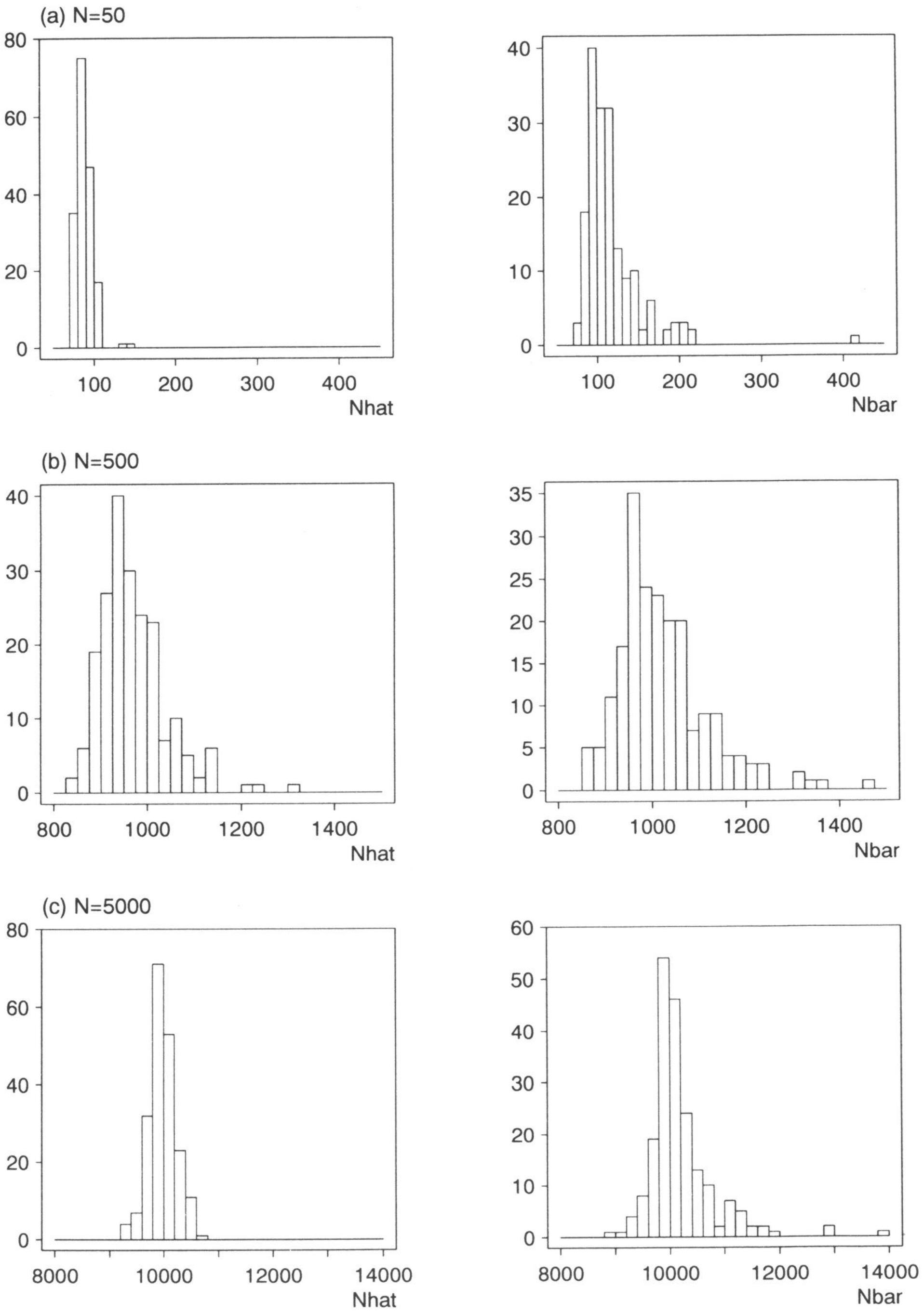


Figure 5.2
Histograms for $\hat{N}$ and $\overline{N}$
(a) N_0=100, (b) N_0=1000, (c) N_0=10000.

From these figures we may draw the following conclusions:

(1) All estimators become better as N_0 gets larger. This is not surprising as we have more statistical information.

(2) α can be estimated relatively well already for small N_0; this in contrary to ε which has for N_0=10000 still a large range.

(3) The value of the likelihood function in θ^{nr} is larger than in $\tilde{\theta}$, and in $\hat{\theta}$ again larger than in θ^{nr}, as might be expected. The differences are in some cases very small, indicating the extreme flatness of the likelihood in the neighbourhood of its maximum.

(4) For almost all data-sets the value of the log-likelihood function at θ^{nr} is already higher than $\log L_0$, the value in the true parameter θ_0.

(5) Different triples can lead to almost same curve (up to τ) and likelihood. The Littlewood class is very broad in the sense that different parameter triples can give a very good fit to the same data. Predictions of events after τ will differ, of course.

(6) In a few cases the likelihood function takes its absolute maximum on the boundary ε=0. We checked that in these cases there are no other (local) maxima. The data here falsely suggests the Jelinski-Moranda model.

Chapter 6

Goodness of Fit*

6.1 Introduction

A classical problem in statistics is the construction of so-called goodness of fit tests, to check whether a model provides a good fit to a given set of data. In this paper we discuss how to attack this problem in the context of statistical models in software reliability. In the classical situation a sequence $x_1, \cdots, x_n$ of independent identically distributed random variables with an unknown distribution function $F(x)$ is given. The problem is to test the null hypothesis that the x_i come from a specified parametric distribution function $F_\theta(t)$, for some $\theta \in \Theta$. The standard way to attack this problem is to compute both $\mathbb{F}_n$, the empirical distribution function of $x_1, \cdots, x_n$, and $F_{\hat\theta}$, where $\hat\theta$ usually denotes the maximum likelihood estimator for θ. The difference $\mathbb{F}_n - F_{\hat\theta}$ should in some sense be small under the null hypothesis, but larger under the alternative hypothesis. For this goodness of fit test the Kolmogorov-Smirnov-test-statistic, which equals

$$T_n := \sup_{x \in \mathbb{R}} \mid \mathbb{F}_n(x) - F_{\hat\theta}(x) \mid ,$$

is often used. A well-known result (see e.g. Shorack & Wellner (1986)) is that for the set Θ consisting of a single point θ_0:

* This chapter is based on the unpublished paper "Goodness of Fit Tests for Statistical Models in Software Reliability" by Mark C. van Pul. A joint paper with Niko O. Maglaperidze and Zura P. Tsigroshvili is in preparation.

$$\sqrt{n}\,(\mathbb{F}_n - F_{\theta_0}) \to_D \mathbb{B}^0 \circ F_{\theta_0},$$

where $\mathbb{B}^0$ denotes the Brownian bridge. Otherwise the large sample distribution can be rather complicated.

We will briefly resume the main concepts in software reliability. For more details we refer to Van Pul (1990). Let $n(t)$, $t \in [0,\tau]$, a counting process, counting the occurrences of software failures up to time t. It's associated intensity function will be denoted by $\lambda(t)$. This stochastic function λ is determined uniquely by the property that it is the predictable function such that

$$m(t) := n(t) - \int_0^t \lambda(s)\mathrm{d}s$$

is a martingale. In this situation we want to test the null hypothesis that the counting process n has an associated intensity function in a certain specified parametric family H_0: $\lambda = \lambda_\theta$, $\theta \in \Theta$ against the alternative hypothesis H_1: $\lambda \neq \lambda_\theta, \theta \in \Theta$. Instead of $\mathbb{F}_n - F_{\hat{\theta}}$ we now look at

$$m(t, \hat{\theta}) := n(t) - a(t, \hat{\theta}),$$

where

$$a(t, \hat{\theta}) = \int_0^t \lambda(s, \hat{\theta})\mathrm{d}s.$$

Here $\hat{\theta}$ denotes the maximum likelihood estimator for θ. In the case that $\Theta = \{\theta_0\}$, that is $H_0{:}a = a_0$ (known), $n a_0^{-1}$ is distributed as a Poisson process, so that a goodness of fit test is quite easy to construct. Another special case is $H_0{:}a = c a_0$ (a_0 known, c unknown) which leads to the total time on test plot (TTT) of n versus a_0. See Gill (1986).

In Section 6.2 we will give some more backgrounds and notation. In the third section we will derive the limit distribution of the Kolmogorov-Smirnov test statistic for the special case of a Poisson process. In Section 6.4 we will investigate the situation for more general counting processes. We will present some numerical results with goodness of fit tests on real data in software reliability and give some concluding remarks.

6.2 The general framework

In order to explain how we will derive large sample distributions, we have to introduce some more notations. Let $n_\nu(t)$ a counting process with intensity of the form

$$\lambda_\nu(t;\theta) = \nu\beta(t;\theta;\nu^{-1} n_\nu)$$

and compensator

$$a_\nu(t;\theta) = \int_0^\tau \lambda_\nu(s,\theta)\mathrm{d}s.$$

We define the stochastic process

$$x_\nu(t) := \nu^{-1} n_\nu(t)$$

and let $x_0(t)$ be the solution of

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = \beta(t;\theta;x), \quad x(0)=0.$$

Furthermore we define

$$m_\nu(t,\theta) := n_\nu(t) - a_\nu(t,\theta), \tag{6.1}$$

$$w_\nu(t,\theta) := \nu^{-1/2} m_\nu(t,\theta),$$

$$h_\nu(t,\theta) := \frac{\partial}{\partial\theta}\log\beta(t;\theta;x_\nu), \tag{6.2}$$

$$h(t,\theta) := \frac{\partial}{\partial\theta}\log\beta(t;\theta;x_0).$$

Suppose $\hat{\theta}_\nu$ is a consistent maximum likelihood estimator of θ. Using a result of Kurtz (1983) and the martingale central limit theorem we can obtain, under reasonable smoothness conditions on β:

$$x_\nu(.) \to_{\mathrm{IP}} x_0(.),$$

$$\nu^{-1/2} m_\nu(.,\theta_0) \to_D w(.), \tag{6.3}$$

$$\nu^{1/2}(\hat{\theta}_\nu - \theta_0) \to_D \left(\int_0^\tau h^{\otimes 2}(u,\theta_0)\mathrm{d}x_0(u)\right)^{-1} \left(\int_0^\tau h(u,\theta_0)\mathrm{d}w(u)\right), \tag{6.4}$$

as $\nu \to \infty$ in respectively $\left(D([0,\tau]), \|.\|\right)$, $\left(D([0,\tau]), J\right)$ and $\left(\mathrm{IR}, d\right)$. Here $h^{\otimes 2} = hh^T$ (i.e. a square matrix with elements $h_i h_j$), w is a zero mean Gaussian martingale with predictable variation process

$$<w>_t := var(w(t)) = x_0(t).$$

Furthermore $D([0,\tau])$ denotes the space of cadlag functions on $[0,\tau]$, $\|.\|$ is the supremum norm in $D([0,\tau])$, J is the Skohorod-metric on $D([0,\tau])$ and d is the Euclidean distance in IR.

It seems reasonable to base a goodness of fit test on the process

$$m_\nu(t,\hat{\theta}_\nu) := n_\nu(t) - a_\nu(t,\hat{\theta}_\nu).$$

By a Taylor-expansion, it can be proved that

$$\nu^{-1/2}\left[a_\nu(t,\hat{\theta}_\nu) - a_\nu(t,\theta_0)\right] = \nu^{1/2}(\hat{\theta}_\nu - \theta_0)^T\int_0^t h(u,\theta_0)\mathrm{d}x_0(u) + O(\nu^{-1/2}). \tag{6.5}$$

Recalling (6.3) and (6.4) and using (6.5) we have

$$\nu^{-1/2}m_\nu(.,\hat{\theta}_\nu) = \nu^{-1/2}\left[n_\nu(.) - a_\nu(.,\theta_0)\right] - \nu^{-1/2}\left[a_\nu(.,\hat{\theta}_\nu) - a_\nu(.,\theta_0)\right]$$

$$\rightarrow_D z(.),$$

where

$$z(t) := w(t) - \left[\int_0^t h^T(u,\theta_0)\mathrm{d}x_0(u)\right]\left[\int_0^\tau h^{\otimes 2}(u,\theta_0)\mathrm{d}x_0(u)\right]^{-1}\left[\int_0^\tau h(u,\theta_0)\mathrm{d}w(u)\right].$$

As $\int_0^\tau h(u,\theta_0)\mathrm{d}w(u)$ is independent of the previous process, as all processes are Gaussian and as

$$\mathrm{cov}\left[z(t)\,,\int_0^\tau h(u,\theta_0)\mathrm{d}w(u)\right] = 0,$$

we can consider

$$w(t) = z(t) + \left[\int_0^t h^T(u,\theta_0)\mathrm{d}x_0(u)\right]\left[\int_0^\tau h^{\otimes 2}(u,\theta_0)\mathrm{d}x_0(u)\right]^{-1}\left[\int_0^\tau h(u,\theta_0)\mathrm{d}w(u)\right]$$

as the decomposition of w as the sum of two independent zero mean Gaussian processes. By Anderson's lemma (1955), we therefore obtain:

$$\mathbb{P}(\,\|w\| \geq a\,) \;\geq\; \mathbb{P}(\,\|z\| \geq a\,);$$

i.e. the Kolmogorov-Smirnov test based on $m_\nu(.,\hat{\theta}_\nu)$, but ignoring the estimation of θ, is conservative.

This leads to two classical approaches:

(i) Just use the Kolmogorov-Smirnov test (i.e. be conservative).
(ii) When testing at the 5%-level, use the 20% critical value as an *ad hoc* correction to conservatism (an idea of A.O. Allen (1978))

We can alternatively make use of two other approaches leading to asymptotically exact tests, avoiding both conservatism (i) and *ad hoc* approximation (ii):

(iii) Compute the distribution of z by simulation or by numerical integration, see Kardaun (1986).
(iv) Transform z and correspondingly z_ν into another process for which the distribution of its supremum norm is easy to find or well known.

In this paper we are going to use approach (iv), following an idea of Khmaladze (1981) in the context of empirical processes. Consider the asymptotic situation and let

$$z^* := z - \mathfrak{z},$$

where $\mathfrak{z}$ is the compensator of z with respect to its natural filtration. In fact, we consider

$$\mathcal{F}_t^z := \sigma\left\{\, w(s),\ s \leq t;\ \int_0^\tau h(u,\theta_0)\mathrm{d}w(u)\,\right\}.$$

It will turn out that z^* is a Gaussian martingale with easy properties. Khmaladze showed that the transformation from z to z^* loses no statistical information, in a certain sense. Clearly $\sigma\{\, z(s): s \leq t\,\} \subseteq \mathcal{F}_t^z$ and equality can also be shown, modulo completion by null sets. One can show that

$$\mathfrak{z}(t) := \int_{s=0}^{t} \left[\; h^T(s,\theta_0)\mathrm{d}x_0(s) \left(\int_{u=s}^{\tau} h^{\otimes 2}(u,\theta_0)\mathrm{d}x_0(u)\right)^{-1} \left(\int_{u=s}^{\tau} h(u,\theta_0)\mathrm{d}w(u)\right) \right]$$

$$- \left(\int_{s=0}^{t} h^T(s,\theta_0)\mathrm{d}x_0(s)\right)\left(\int_{s=0}^{\tau} h^{\otimes 2}(u,\theta_0)\mathrm{d}x_0(u)\right)^{-1}\left(\int_{s=0}^{\tau} h(u,\theta_0)\mathrm{d}w(u)\right) \qquad (6.6)$$

simply by verifying that

$$z^*(t) = w(t) - \int_{s=0}^{t} \left[h^T(s,\theta_0)\mathrm{d}x_0(s) \left(\int_{u=s}^{\tau} h^{\otimes 2}(u,\theta_0)\mathrm{d}x_0(u) \right)^{-1} \left(\int_{u=s}^{\tau} h(u,\theta_0)\mathrm{d}w(u) \right) \right]$$

is indeed a martingale. So if we can construct a transform of $\nu^{-1/2}m_\nu(.,\hat{\theta}_\nu)$ which is asymptotically distributed as $z^*(.)$, this is in fact the asymptotic distribution of $\nu^{-1/2}m_\nu(.,\theta_0)$; thus after transformation we may entirely forget the estimation of θ_0. The most obvious transformation seems to be the following. We simply substitute $h_\nu(t,\hat{\theta}_\nu)$, $x_\nu(t)$ and $\nu^{-1/2}m_\nu(t,\hat{\theta}_\nu)$ for $h(t,\theta_0)$, $x_0(t)$ and $w(t)$ in (6.6). The substitution of the first two estimates seems harmless, the last one is more subtle. In stead of $\nu^{-1/2}m_\nu(.,\hat{\theta}_\nu)$, we should really want to substitute $\nu^{-1/2}m_\nu(.,\theta_0)$ for $w(.)$, because of (6.3). Their difference, however, is asymptotically given by (6.5). Replacing w in (6.6) by the right hand side of (6.5) gives zero. In other words, the mapping

$$w \rightarrow w - \tilde{w}$$

is a projection. So Geurts *et al.* (1988) conclude that one can probably use the goodness of fit test statistic $T_\nu(\tau)$ given by

$$\frac{\sup\limits_{t\in[0,\tau]} \left| m_\nu(t,\hat{\theta}_\nu) - \int_{s=0}^{t} h_\nu^T(s,\hat{\theta}_\nu)dx_\nu(s) \left(\int_{u=s}^{\tau} h_\nu^{\otimes 2}(u,\hat{\theta}_\nu)\mathrm{d}x_\nu(u) \right)^{-1} \left(\int_{u=s}^{\tau} h_\nu(u,\hat{\theta}_\nu)\mathrm{d}m_\nu(u,\hat{\theta}_\nu) \right) \right|}{\left(\nu x_\nu(\tau) \right)^{1/2}} \tag{6.7}$$

which is expected to be asymptotically distributed as

$$\sup_{t\in[0,1]} |I\!B(t)|,$$

where $I\!B$ denotes the standard Brownian motion. More precisely, they stated the following conjecture:

Conjecture 6.1 *Let be given a counting process $n_\nu(t)$ with underlying monotone non-increasing intensity function $\lambda_\nu(t;\theta)$. We define the martingale $m_\nu(t;\theta)$, the score function $h_\nu(t;\theta)$ and Kolmogorov-Smirnov test-statistic $T_\nu(\tau)$ in the usual way (c.f. (6.1), (6.2) and (6.7)). Then we have as $\nu\rightarrow\infty$:*

$$T_\nu(\tau) \rightarrow_D \sup_{t\in[0,1]} |I\!B(t)|, \tag{6.8}$$

where $I\!B$ denotes the standard Brownian motion.

Remark 6.1 A serious problem could arise, in our opinion, as in formula (6.7)

$$\int_{u=s}^{\tau} h_{\nu}^{\otimes 2}(u,\hat{\theta}_{\nu})\mathrm{d}x_{\nu}(u)$$

becomes very small (tends to 0) for s converging to τ. Because of this, it is not obvious at all that the messy integral in (6.7) does not explode for t close to τ. Therefore we have to study the behaviour of $T_{\nu}(t)$ for t close to τ carefully. Let us reconsider more closely the integral in the right hand side of expression (6.7):

$$\int_{s=0}^{t} h^{T}(s)\left[\int_{u=s}^{\tau} h^{\otimes 2}(u)\mathrm{d}n(u)\right]^{-1}\left[\int_{u=s}^{\tau} h(u)\mathrm{d}m(u)\right]\mathrm{d}n(s)$$

$$= \sum_{i=1}^{n(t)} h^{T}(T_i)\left[\int_{u=T_i}^{\tau} h^{\otimes 2}(u)\mathrm{d}n(u)\right]^{-1}\int_{u=T_i}^{\tau} h(u)\mathrm{d}m(u).$$

Notice that if $t \geq T_{n(\tau)}$ the term with index $i = n(\tau)$ contains the inverse of

$$\int_{u=T_{n(\tau)}}^{\tau} h^{\otimes 2}(u)\mathrm{d}n(u) = h^{\otimes 2}(T_{n(\tau)}).$$

We see immediately that if h is more than one-dimensional, this will lead to explosion because of the zeroness of the determinant. Therefore, in the general case (6.7) is only defined for $t < T_{n(\tau)}$. This contradicts Conjecture 6.1. In the sequel of this chapter we will investigate whether the conjecture holds if the supremum in the test statistic (6.7) is taken over $[0, T_{n(\tau)})$. □

Remark 6.2 Because $<z^{*}>=<w>=x_0$, it follows for all $\varepsilon>0$:

$$\sqrt{\frac{n_{\nu}(\tau-\varepsilon)}{n_{\nu}(\tau)}}\,T_{\nu}(\tau-\varepsilon) \rightarrow_{D} \sup_{t\in[0,\frac{n_{\nu}(\tau-\varepsilon)}{n_{\nu}(\tau)}]} |\mathbb{B}(t)|, \tag{6.9}$$

and less elegantly

$$T_{\nu}(\tau-\varepsilon) \rightarrow_{D} \sup_{t\in[0,1]} |\mathbb{B}(t)|.$$

Because of (6.9), to prove (6.8) it is sufficient to show that for all $\delta>0$:

$$\lim_{\varepsilon\downarrow 0}\ \limsup_{\nu\to\infty}\ \mathbb{P}\left(\sup_{t\in[\tau-\varepsilon,\tau]} \frac{|A_\nu(t)-A_\nu(\tau-\varepsilon)|}{n_\nu(\tau)} > \delta \right) = 0, \tag{6.10}$$

where

$$A_\nu(t) \ := \ m_\nu(t,\hat\theta_\nu) - \int_{s=0}^{t} \left[h^T \mathrm{d}n_\nu \left[\int_{u=s}^{\tau} h^{\otimes 2}\mathrm{d}n_\nu \right]^{-1} \left[\int_{u=s}^{\tau} h\,\mathrm{d}m_\nu \right] \right]. \tag{6.11}$$

Note that $A_\nu(t)$ only exists for $t < \tau$ (see also Khmaladze (1983)). For test statistics as the one in (6.9) a fixed fraction of the information is not used. We are more interested in the limit distribution of test statistics for which the fraction of information lost tends to zero as ν grows. □

6.3 The Poisson case

In this section we will prove Conjecture 6.1 for the special case of a Poisson process. Note that the parameter θ in in this case one-dimensional (see also Remark 6.1). The following result for Poisson processes will be very useful.

Theorem 6.1 *Let $n_\nu(t)$ be a counting process with a constant failure intensity $\lambda_\nu(t)=\nu\lambda$, observed on a finite time interval $[0,\tau]$. We define*

$$S_\nu(\varepsilon) \ := \ \sup_{t\in[0,\varepsilon]} \nu^{1/2} \left| \int_t^{\varepsilon} \frac{s}{n_\nu(s)}\mathrm{d}n_\nu(s) \ - \ (\varepsilon - t) \right|, \quad \varepsilon\in[0,\tau].$$

Then for all $\delta>0$ we have

$$\lim_{\varepsilon\downarrow 0}\ \limsup_{\nu\to\infty}\ \mathbb{P}\,(\, S_\nu(\varepsilon) > \delta\,) \ = \ 0. \tag{6.12}$$

Proof Theorem 6.1 We note that for Poisson processes $n_\nu(t),\ t\in[0,\tau]$:

$$\frac{n_\nu(t)}{n_\nu(\tau)}\Big|_{n_\nu(\tau)=n} \ \sim \ \mathbb{H}_n(t), \tag{6.13}$$

where $\mathbb{H}_n$ denotes the empirical distribution function of n uniform random variables in $[0,\tau]$. Notice that for any function f:

$$\left|\int_t^{\varepsilon} f(u)\mathrm{d}u\right| \ \le \ \left|\int_0^{\varepsilon} f(u)\mathrm{d}u\right| \ + \ \left|\int_0^{t} f(u)\mathrm{d}u\right|$$

and hence

$$\sup_{t\in[0,\varepsilon]}\left|\int_t^{\varepsilon} f(u)\mathrm{d}u\right| \le 2 \sup_{t\in[0,\varepsilon]}\left|\int_0^t f(u)\mathrm{d}u\right|. \tag{6.14}$$

Conditioning on $n_\nu(\tau)=n$, it follows from (6.13) and (6.14) that:

$$\begin{aligned}
0 &\le \frac{1}{2}S_\nu(\varepsilon)\\
&\le \sup_{t\in[0,\varepsilon]} \nu^{1/2}\left|\int_0^t \frac{s}{n_\nu(s)}\mathrm{d}n_\nu(s) - t\right|\\
&\le \sup_{t\in[0,\varepsilon]} \nu^{1/2}\left|\frac{n_\nu(t)}{n_\nu(\tau)} - t\right| + \sup_{t\in[0,\varepsilon]}\left|\int_0^t \frac{\nu^{1/2}\left(s - \dfrac{n_\nu(s)}{n_\nu(\tau)}\right)}{\dfrac{n_\nu(s)}{n_\nu(\tau)}}\mathrm{d}\frac{n_\nu(s)}{n_\nu(\tau)}\right|\\
&\sim \sup_{t\in[0,\varepsilon]} \sqrt{n}\left|I\!H_n(t) - t\right| + \sup_{t\in[0,\varepsilon]}\left|\int_0^t \frac{\sqrt{n}(s - I\!H_n(s))}{I\!H_n(s)}\mathrm{d}I\!H_n(s)\right|\\
&= \sup_{t\in[0,\varepsilon]}|I\!K_n(t)| + \sup_{t\in[0,\varepsilon]}|I\!M_n(t)|,
\end{aligned} \tag{6.15}$$

with

$$I\!K_n(t) := \sqrt{n}\left(I\!H_n(t) - t\right)$$

and

$$I\!M_n(t) := \int_0^t \frac{\sqrt{n}(s - I\!H_n(s))}{I\!H_n(s)}\mathrm{d}I\!H_n(s). \tag{6.16}$$

From a well-known theorem in the theory of empirical processes (see e.g. Shorack & Wellner (1986), p. 110) it follows that:

$$\sup_{t\in[0,\varepsilon]} |I\!K_n(t)| \to_D \sup_{t\in[0,\varepsilon]} |I\!B^0(t)|,$$

where $I\!B^0$ denotes the Brownian bridge. Because the Brownian bridge has continuous paths and because $I\!B^0(0)=0$, we have for all $\delta>0$:

$$\lim_{\varepsilon\downarrow 0} \limsup_{n\to\infty} I\!P(\sup_{t\in[0,\varepsilon]} |I\!K_n(t)|>\delta) = 0. \tag{6.17}$$

If we can prove also (6.17) with $I\!K_n$ replaced by $I\!M_n$, then from (6.15) the result (6.12) of Theorem 6.1 will follow immediately. We will need the following lemma:

Lemma 6.1 *With $I\!M_n$ given by (6.16) we have*

$$\sup_{t\in[0,1]} |I\!M_n(t) - I\!M(t)| \overset{a.s.}{\to} 0,$$

where

$$I\!M(t) := \int_0^t \frac{I\!K(s)}{s} \mathrm{d}s, \quad s\in[0,1].$$

Proof Lemma 6.1 The proof of Lemma 6.1, which is technical but routine, is omitted here. □

Because taking the supremum over a finite interval is a continuous and measurable operator, the continuous mapping theorem gives

$$\sup_{t\in[0,\varepsilon]} |I\!M_n(t)| \to_D \sup_{t\in[0,\varepsilon]} \left| \int_0^t \frac{I\!B^0(s)}{s} \mathrm{d}s \right|$$

As the right hand side of (6.4) has continuous paths and is zero in zero we get that for all $\delta>0$:

$$\lim_{\varepsilon\downarrow 0} \limsup_{n\to\infty} I\!P(I\!M_n(\varepsilon)>\delta) = 0. \tag{6.18}$$

Now from (6.15), (6.17) and (6.18) the result (6.12) of Theorem 6.1 will follow immediately. □

We are now able to prove the following result:

Theorem 6.2 *Let be given a Poisson counting process $n_\nu(t)$ with constant intensity function θ. Then the result (6.8) of Conjecture 6.1 holds.*

Proof Theorem 6.2 We will derive (6.10) with A_ν in (6.11) defined for a Poisson process. For a Poisson process with parameter θ expression (6.11) becomes

$$A_\nu(t) = n_\nu - \nu\theta t - \int_0^t \frac{m_\nu(\tau) - m_\nu(s)}{n_\nu(\tau) - n_\nu(s)} \, dn_\nu(s)$$

$$= \theta \left[\int_0^t \frac{\tau - s}{n_\nu(\tau) - n_\nu(s)} \, dn_\nu(s) - t \right]$$

and

$$\frac{|A_\nu(t) - A_\nu(\tau-\varepsilon)|}{\sqrt{n_\nu(\tau)}} = \frac{\theta}{\sqrt{n_\nu(\tau)}} \left| \int_{\tau-\varepsilon}^t \frac{\tau - s}{n_\nu(\tau) - n_\nu(s)} dn_\nu(s) - t + (\tau-\varepsilon) \right|.$$

Because

$$\frac{\nu\theta}{\sqrt{\nu n_\nu(\tau)}} \rightarrow_{\mathrm{IP}} M,$$

for some finite constant M, we concentrate our intention on

$$T_\nu(\varepsilon) := \sup_{t\in[\tau-\varepsilon,\tau]} \nu^{1/2} \left| \int_{\tau-\varepsilon}^t \frac{\tau - s}{n_\nu(\tau) - n_\nu(s)} dn_\nu(s) - t + (\tau-\varepsilon) \right|$$

$$\sim \sup_{t\in[0,\varepsilon]} \nu^{1/2} \left| \int_t^\varepsilon \frac{s}{n_\nu(s)} dn_\nu(s) - (\varepsilon-t) \right|.$$

Applying now Theorem 6.1 will give us the desired result. This proves Theorem 6.2. □

Remark 6.3 It is not difficult to see that the result of Theorem 6.2 can easily generalised to non-homogeneous Poisson counting processes with intensity $\theta(t)$ by an appropriate inverse time transformation. It is, however, not clear to us how this result could be proved for more general arbitrary counting processes with monotone non-increasing intensity function $\lambda(t,\theta)$. This is perhaps possible by studying the homogeneous Poisson process with parameter $\lambda(\tau,\theta)$. In case θ is more-dimensional, we have to consider the adapted version of Conjecture 6.1, that is with the supremum of the test statistic (6.7) taken over $[0, T_{n(\tau)})$.

This problem is still open too. Simulations have shown, however, that if 6.8 holds also for other than Poisson processes, convergence to the limit distribution can be rather slow. See Example 6.2 in the next section. □

6.4 Some numerical examples

We will now investigate how slightly adapted test statistics will behave in practice. As stated before, for a test statistic of the form $T(\tau-\varepsilon)$ (i.e. a fixed fraction of the information is not used) it can be easily shown that it will converge to the same limit distribution as is mentioned in (6.8). This in contrary to test statistics of the form $T(T_{n(\tau)-k})$ (where the fraction of information lost diminishes as N_0 tends to infinity) for which we could only prove Conjecture 6.1 in the special case of a Poisson process.

Example 6.1 *A case study.* For the data of Moek's project A (see also Example 3.3) we found 2.22 for the value of the test statistic $T(T_{n(\tau)}-)$ according to the Jelinski-Moranda model. The critical value, at the 95 percent level, equals 2.25. So we have to conclude that Moek's data do not reject the Jelinski-Moranda model. □

N_0	T_1	T_2	T_3
100	46	74	68
200	64	79	85
300	68	82	87
400	70	84	89
500	71	79	85
600	76	85	92
700	78	82	90
800	70	78	87
900	81	90	95

Table 6.1
Percentages of test statistics T_1, T_2 and T_3 lower than critical value at 95% level.

Example 6.2 *Simulation of the Jelinski-Moranda model.* We simulated the Jelinski-Moranda model with N_0=100, ϕ_0=1 and τ=1. For various values of N_0 we computed 100 replicates of test statistics $T_1:=T(T_{n(\tau)}-)$, $T_2:=T(T_{n(\tau)-5})$ and $T_3:=T(0.9\tau)$. The quantiles that lay under the 95 percent critical value are given in Table 6.1. The figures indicate that the convergence of the test statistic to its limit distribution is extremely slow. This puts, of course, serious limitations to the usefulness of Khmaladze's approach, as described in Section 6.2. □

Remark 6.4 An other idea is to substitute in the test statistic (6.7) $x_0(t;\hat{\theta}_\nu)$ for $x_0(t)$ (instead of $x_\nu(t)$). The test statistic is then well defined on $[0,\tau]$ and Maglaperidze & Tsigroshvili (1993) showed that under suitable conditions Conjecture 6.1 holds for the resulting test statistic. A disadvantage of this approach is that the actual computations become a lot more complicated since instead of finite sums, integrals have to be calculated. This seems to contradict Khmaladze's original idea, which was to find a test statistic which was easy to use in practice. A second disadvantage is that Maglaperidze & Tsigroshvili's conditions also concern these theoretical integrals and therefore it is not obvious in advance whether they will hold or not. This means that for each different model still a lot of work has to be done. It would be interesting to know, nevertheless, whether this approach will give a faster convergence to the limit distribution. See Maglaperidze *et al.* (1993). □

Chapter 7

Modelling Imperfect Repair and Software Growth*

7.1 Introduction

The reliability of hard- and software is in some applications of vital importance to their users. During the recent Gulf war a patriot missile, which was stationed in Turkey, was fired by accident, because of a bug in the software. Obviously also in case of less delicate computer applications customers want a high degree of reliability to be guaranteed. The modelling of the evolution of the reliability of a piece of software undergoing debugging will be the subject of this paper.

In software reliability theory many different models have been proposed and investigated. Most of these models assume perfect repair and constant software size. Both restrictions oversimplify reality in a huge way. In the model we discuss in this paper, we have tried to overcome both simplifications in such a way that statistical inference is still possible.

In the next section we give some backgrounds and classical assumptions of software reliability theory. In Section 7.3 we describe the PG&IR model, a new model with interesting features, and we suggest how to estimate the model parameters. In the fourth section we discuss the statistical properties of this model. In Section 7.5 we shed some light upon the huge amount of extensions, that are possible, starting from this model; we define a class of regression models. Finally, in the sixth and last section we give some concluding remarks.

* This chapter is based on the paper "Statistical Models in Software Reliability" by Mark C. van Pul, which appeared in the *Proceedings of the 23rd. Symposium on the Interface* (1991), pp. 106-109.

7.2 Backgrounds and classical assumptions

Let us consider the following test experiment. A very large computer program is executed during a fixed exposure period, say $[0,\tau]$. Inputs are selected 'at random' from the input space, that is, they are generated in such a way that they are representative for the operational profile. For each input the program either produces the correct output or a software failure is detected; the software produces a wrong answer or no answer at all. After the detection of a failure the CPU-clock is stopped and the software is sent to a team of debuggers. The failure time and possibly other failure data are observed. After the bug is found and fixed, the CPU-clock is restarted again and testing continues with a new input until time τ is reached.

Efforts in describing the evolution of the reliability of computer software during test and development resulted in the proposal of dozens of new models over the past twenty years. An important class of such models is the so-called class of Error-Counting and Debugging (EC&D) models. This class consists of models that are based on the test experiment described above (with only the failure times as test data) and some strong assumptions:

(A1) *Perfect repair:* no new faults are introduced during a repair with probability 1.
(A2) *Fixed software size:* there is no addition of new software during testing.
(A3) *Independence of faults:* faults (and hence their failure times) are independent.

Although all three assumptions seem to be rather unrealistic, they form a framework on which many models are built. The most elementary and oldest software reliability model is the model of Jelinski-Moranda (1972), introduced almost twenty years ago. In this model the failure rate of the program is assumed to be at any time proportional to the number of remaining faults and the repair of each fault does make the same contribution to the decrease in failure rate. Denoting $n(t)$ for the observed counting process, we find for the failure intensity function $\lambda(t)$ the following expression:

$$\lambda(t) \;=\; \phi\left(N - n(t-)\right), \quad t \in [0,\tau],$$

with model parameters N, the number of faults initially present in the software, and ϕ, the occurrence rate per fault, which can also be interpreted as the test efficiency. Musa (1975), Littlewood (1980) and many others have built more sophisticated models, for technical reasons, however, generally restricted by assumptions (A1)-(A3).

As there exist no perfect testers and programmers, there will always be a positive chance of introducing new faults, while repairing an old one. Secondly, development and testing of software usually takes place simultaneously in practice. Because the addition of software, that has never been tested before, will have an effect on the reliability, it seems reasonable to take also software growth during testing into account. Furthermore certain bugs will prevent parts of the software to be inspected and therefore will hide other bugs, thus violating the assumption of independence of faults. Dropping (A3), however, would cause the mathematical problem to become highly complicated and almost untractable.

In the next section we introduce a new model, the Poisson Growth and Imperfect Repair (PG&IR) model. We combined the modelling of imperfect repair and software growth in a natural way. Furthermore to a certain extent the model will account for dependencies between faults. The model has attractive statistical properties, besides.

7.3 The PG&IR model

Let $\tau>0$. We consider a test experiment as described in the introduction. Let $T_0:=0$ and $T_i, i=1,2,...$ the failure times of the occurring failures. Repair takes place immediately after a failure is detected. After each repair the addition of new software is allowed. Due to the correction of a fault and eventually due to the addition of new software at time T_i, there is a change in software of size $K_i, i=0,1,....$ We assume that at time T_i apart from deleting one fault, N_i new faults are introduced. We assume that N_i is a stochastic variable, Poisson distributed with mean $\mu K_i, i=0,1,....$ We consider the testing process during $[0,\tau]$, observing say $n(\tau)$ faults. Let

$$n(t) := \sum_{i=1}^{n(\tau)} I\{T_i < t\}, \qquad t \in [0,\tau], \tag{7.1}$$

the number of failures detected (faults deleted) during $[0,t]$ and let

$$N(t) := \sum_{i=0}^{n(\tau)} N_i I\{T_i < t\}, \qquad t \in [0,\tau], \tag{7.2}$$

the number of faults introduced during $[0,t]$. We assume that the failure intensity λ, like in the Jelinski-Moranda model (1972), at any time is proportional to the remaining number of faults, that is:

$$\lambda(t) := \phi[N(t-)-n(t-)], \qquad t \in [0,\tau], \tag{7.3}$$

where ϕ denotes the constant occurrence rate per fault. With use of the data (T_i,K_i), $i=0,1,...,n(\tau)$, obtained from the experiment as described above, one can estimate the parameters (μ,ϕ) of the underlying PG&IR model. We use the maximum likelihood estimation (MLE) procedure for this purpose. The following lemma will be very useful:

Lemma 7.1 *For all $m \in \mathrm{IN}$ and all $(a_0,a_1, \cdots a_m) \in \mathrm{IR}_+^{m+1}$, we have*

$$\sum_{N_0=0}^{\infty} N_0 \frac{a_0^{N_0}}{N_0!} \sum_{N_1=0}^{\infty} (N_0+N_1-1)\frac{a_1^{N_1}}{N_1!} \cdots \sum_{N_m=0}^{\infty} (N_0+N_1+ \cdots +N_m-m)\frac{a_m^{N_m}}{N_m!}$$
$$= a_0(a_0+a_1) \cdots (a_0+a_1+ \cdots +a_m)\, e^{a_0+a_1+ \cdots +a_m}. \tag{7.4}$$

Proof Lemma 7.1 Let $m:=0$ and $a_0 \in \mathrm{IR}_+$. Then we have

$$\sum_{N_0}^{\infty} N_0 \frac{a_0^{N_0}}{N_0!} = a_0 \sum_{N_0=1}^{\infty} \frac{a_0^{N_0-1}}{(N_0-1)!}$$
$$= a_0 \sum_{N_0=0}^{\infty} \frac{a_0^{N_0}}{N_0!}$$
$$= a_0 e^{a_0}.$$

Suppose we have proved the induction hypothesis (7.4) for a certain $m-1 \in \mathbb{N}$ and for all $(a_0,a_1,\cdots,a_{m-1}) \in \mathbb{R}_+^m$. Then we have for all $(a_0,a_1,\cdots,a_m) \in \mathbb{R}_+^{m+1}$:

$$\sum_{N_0=0}^{\infty} N_0 \frac{a_0^{N_0}}{N_0!} \sum_{N_1=0}^{\infty} (N_0+N_1-1)\frac{a_1^{N_1}}{N_1!}$$

$$\times \sum_{N_2=0}^{\infty} (N_0+N_1+N_2-2)\frac{a_2^{N_2}}{N_2!} \quad \cdots \quad \sum_{N_m=0}^{\infty} (N_0+N_1+\cdots+N_m-m)\frac{a_m^{N_m}}{N_m!} \quad =$$

$$= a_0 \sum_{N_0=1}^{\infty} \sum_{N_1=0}^{\infty} (N_0-1+N_1)\frac{a_0^{N_0-1}}{(N_0-1)!}\frac{a_1^{N_1}}{N_1!}$$

$$\times \sum_{N_2=0}^{\infty} (N_0-1+N_1+N_2-1)\frac{a_2^{N_2}}{N_2!} \quad \cdots \quad \sum_{N_m=0}^{\infty} (N_0-1+N_1+\cdots+N_m-(m-1))\frac{a_m^{N_m}}{N_m!}$$

$$= a_0 \sum_{N_0=0}^{\infty} \sum_{N_1=0}^{\infty} \frac{N_0+N_1}{(N_0+N_1)!} \binom{N_0+N_1}{N_0} a_0^{N_0} a_1^{N_1}$$

$$\times \sum_{N_2=0}^{\infty} (N_0+N_1+N_2-1)\frac{a_2^{N_2}}{N_2!} \quad \cdots \quad \sum_{N_m=0}^{\infty} (N_0+N_1+\cdots+N_m-(m-1))\frac{a_m^{N_m}}{N_m!}$$

$$= a_0 \sum_{S=0}^{\infty} \sum_{L=0}^{S} \frac{S}{S!} \binom{S}{L} a_0^L a_1^S$$

$$\times \sum_{N_2=0}^{\infty} (S+N_2-1)\frac{a_2^{N_2}}{N_2!} \quad \cdots \quad \sum_{N_m=0}^{\infty} (S+N_2+\cdots+N_m-(m-1))\frac{a_m^{N_m}}{N_m!}$$

$$= a_0 \sum_{S=0}^{\infty} S \frac{(a_0+a_1^{S})}{S!} \sum_{N_2=0}^{\infty} (S+N_2-1)\frac{a_2^{N_2}}{N_2!} \quad \cdots \quad \sum_{N_m=0}^{\infty} (S+N_2+\cdots+N_m-(m-1))\frac{a_m^{N_m}}{N_m!}$$

$$= a_0\,(a_0+a_1) \cdots (a_0+a_1+\cdots+a_m)\,\mathrm{e}^{a_0+a_1+\cdots+a_m}.$$

We reorder the terms of the first and second sum in such a way that an appropriate transformation $(N_0,N_1) \rightarrow (S,L)$ is possible. Then we apply Newton's binomium. Finally, we use the assumption that the induction hypothesis (7.4) is true for $m-1$. This proves the lemma. □

We now return to the derivation of the likelihood function for the PG&IR model, as described by (7.1)-(7.3). Aalen (1978) showed, that the likelihood function for estimating the parameters of the intensity function of a counting process, observed on a fixed time interval $[0,\tau]$ is given by:

$$L_\tau(\mu,\phi;T_0,T_1,\cdots,T_{n(\tau)}) \;=\; \prod_{i=1}^{n(\tau)} \lambda(T_i-)\ \exp(-\int_0^\tau \lambda(s)ds).$$

As the N_i are independent Poisson distributed stochastic variables with mean μK_i, and defining:

$$a_i \;:=\; \mu K_i e^{-\phi(\tau-T_i)}, \tag{7.5}$$

$$b_i \;:=\; (N_0+\cdots+N_{i-1}=i), \tag{7.6}$$

for $i=1,\ldots,n(\tau)$, we get for the likelihood function under the finer filtration (observing also the sizes of the software changes):

$$L_\tau(\mu,\phi;(T_i,K_i),i=0,1,\ldots,n(\tau)) \;=$$

$$= \sum_{N_0=1}^{\infty}\sum_{N_1=b_1}^{\infty}\cdots\sum_{N_{n(\tau)-1}=b_{n(\tau)-1}}^{\infty}\sum_{N_{n(\tau)}=0}^{\infty}\prod_{i=0}^{n(\tau)}\left[\frac{(\mu K_i)^{N_i}}{(N_i)!}e^{-\mu K_i}\right]\prod_{i=1}^{n(\tau)}\phi(\sum_{j=0}^{i-1}N_j-i+1)e^{-\phi\left[\sum_{i=0}^{n(\tau)}(\tau-T_i)N_i+\sum_{i=1}^{n(\tau)}T_i-\tau n(\tau)\right]}$$

$$= \phi^{n(\tau)}\ \exp\left[\phi\sum_{i=1}^{n(\tau)}(\tau-T_i)-\mu\sum_{i=0}^{n(\tau)}K_i\right]$$

$$\times \sum_{N_0=0}^{\infty}N_0\frac{a_0{}^{N_0}}{N_0!}\sum_{N_1=0}^{\infty}(N_0+N_1-1)\frac{a_1{}^{N_1}}{N_1!}\ \cdots\ \sum_{N_{n(\tau)}=0}^{\infty}(N_0+N_1+\ \cdots\ +N_{n(\tau)}-n(\tau))\frac{a_{n(\tau)}{}^{N_{n(\tau)}}}{N_{n(\tau)}!}$$

$$= \phi^{n(\tau)}\ \exp\left[\phi\sum_{i=1}^{n(\tau)}(\tau-T_i)-\mu\sum_{i=0}^{n(\tau)}K_i\right]\ \sum_{i=0}^{n(\tau)}a_i\ \prod_{i=0}^{n(\tau)-1}\left[\sum_{j=0}^{i}a_j\right]$$

$$= (\mu\phi)^{n(\tau)}\ \exp\left[\phi\sum_{i=1}^{n(\tau)}(\tau-T_i)-\mu\sum_{i=0}^{n(\tau)}K_i(1-e^{-\phi(\tau-T_i)})\right]\ \prod_{i=0}^{n(\tau)-1}\left[\sum_{j=0}^{i}K_je^{-\phi(\tau-T_j)}\right]. \tag{7.7}$$

We note that if $N_0+\cdots+N_{i-1}=1$, that is, if $b_i=1$ (so we have to sum N_i from 1 to ∞), then the coefficient $(N_0+\ldots+N_i-i)$ in the i-th sum equals zero for $N_i=0$. So we can take all lower-bounds equal to zero and use Lemma 7.1 to get (7.7).

Taking the logarithm of the likelihood function (7.7) yields

$$\log L_\tau(\mu,\phi) = n(\tau)\log(\mu\phi) - \mu \sum_{i=0}^{n(\tau)} K_i \left(1-e^{-\phi(\tau-T_i)}\right)$$

$$+ \phi \sum_{i=1}^{n(\tau)} (\tau-T_i) + \sum_{i=0}^{n(\tau)-1} \log\left[\sum_{j=0}^{i} K_j e^{-\phi(\tau-T_j)}\right].$$

So we find the likelihood equations:

$$0 = \frac{\partial}{\partial\mu}\log L_\tau(\mu,\phi)$$

$$= \frac{n(\tau)}{\mu} - \sum_{i=0}^{n(\tau)} K_i\left(1-e^{-\phi(\tau-T_i)}\right), \tag{7.8}$$

$$0 = \frac{\partial}{\partial\phi}\log L_\tau(\mu,\phi)$$

$$= \frac{n(\tau)}{\phi} - \mu \sum_{i=0}^{n(\tau)} K_i(\tau-T_i)e^{-\phi(\tau-T_i)} + \sum_{i=1}^{n(\tau)}(\tau-T_i) - \sum_{i=0}^{n(\tau)-1}\left[\frac{\sum_{j=0}^{i} K_j(\tau-T_j)e^{-\phi(\tau-T_j)}}{\sum_{j=0}^{i} K_j e^{-\phi(\tau-T_j)}}\right]. \tag{7.9}$$

Solving the system of equations (7.8)-(7.9), we get the maximum likelihood estimators $(\hat{\mu}.\hat{\phi})$, where

$$\hat{\mu} := \frac{n(\tau)}{\sum_{i=0}^{n(\tau)} K_i\left(1-e^{-\hat{\phi}(\tau-T_i)}\right)}$$

and $\hat{\phi}$ the solution of $g(\hat{\phi})=0$ with

$$g(\phi) := \frac{1}{n(\tau)}\sum_{i=1}^{n(\tau)}(\tau-T_i) + \frac{1}{\phi}$$

$$- \frac{\sum_{i=0}^{n(\tau)} K_i(\tau-T_i)e^{-\phi(\tau-T_i)}}{\sum_{i=0}^{n(\tau)} K_i\left(1-e^{-\phi(\tau-T_i)}\right)} - \frac{1}{n(\tau)}\sum_{i=0}^{n(\tau)-1}\left[\frac{\sum_{j=0}^{i} K_j(\tau-T_j)e^{-\phi(\tau-T_j)}}{\sum_{j=0}^{i} K_j e^{-\phi(\tau-T_j)}}\right].$$

7.4 Asymptotic properties of the MLE's

In Van Pul (1990) it is shown that for a general class of models with intensity functions of the form

$$\lambda_\nu(t,\theta) = \nu\beta(t,\theta,\frac{n_\nu}{\nu}), \qquad t\in[0,\tau], \quad \theta\in\Theta\subset \mathrm{IR}^p, \tag{7.10}$$

consistency, asymptotic normality and efficiency of the maximum likelihood estimators for the model parameters can be proved under certain conditions on β.

In this section we show that if we apply asymptotic theory to the PG&IR model in a natural way, the intensity function can be written in the special form (7.10) and that the corresponding β satisfies the conditions (GC1)-(GC2)&(LC1)-(LC3), given in Theorem 1 of Van Pul (1990).

Let us consider the PG&IR model as given by (7.1)-(7.3). Note that the process $N(t)$ is unobservable. Thus defining the filtrations

$$\mathcal{F}_{t-} := \{\, n(s) : 0 \le s < t \,\},$$

$$\mathcal{G}_{t-} := \{\, n(s), N(s) : 0 \le s < t \,\},$$

we notice that the intensity λ given in (7.3) is actually $\lambda^{\mathcal{G}}$, the intensity function of the counting process with respect to the filtration $\mathcal{G}_t$. With use of the Innovation Theorem (see Aalen (1978) or Bremaud (1977)), we get

$$\lambda^{\mathcal{F}}(t) := \mathrm{IE}\left[\lambda^{\mathcal{G}}(t) \mid \mathcal{F}_{t-}\right]$$

$$= \mathrm{IE}\left[\,\phi[N(t-)-n(t-)] \mid \mathcal{F}_{t-}\right]$$

$$= \phi\left[\mathrm{IE}\,(\,N(t-) \mid \mathcal{F}_{t-}\,) - n(t-)\right]. \tag{7.11}$$

In order to compute

$$\mathrm{IE}\,(\,N(t-) \mid \mathcal{F}_{t-}\,) =$$

$$= \sum_{N_0=0}^{\infty} \cdots \sum_{N_{n(t-)}=0}^{\infty} (N_0 + \cdots + N_{n(t-)})\, f\,(N_0,\ldots,N_{n(t-)} \mid n(t-)), \tag{7.12}$$

recall that

$$f(\vec{N}) = f(N_1, \cdots, N_n)$$

$$= \prod_{i=0}^{n} \left[\frac{(\mu K_i)^{N_i}}{(N_i)!} e^{-\mu K_i} \right] \tag{7.13}$$

and note that the density functions $f(\vec{N},n)$ and $f(n)$ now can be deduced from (7.13) and (7.5):

$$f(\vec{N},n) = \prod_{i=0}^{n} \left[\frac{(\mu K_i)^{N_i}}{(N_i)!} e^{-\mu K_i} \right] \prod_{i=1}^{n} \phi\left(\sum_{j=0}^{i-1} N_j - i + 1 \right) e^{-\phi \left[\sum_{i=0}^{n} (\tau - T_i) N_i + \sum_{i=1}^{n} T_i - \tau n \right]}$$

$$= \phi^n \exp\left[\phi \sum_{i=1}^{n} (\tau - T_i) - \mu \Sigma K_i \right] \prod_{i=0}^{n} \left[\left(\sum_{j=0}^{i-1} N_j - i + 1 \right) \frac{\left[\mu K_i e^{-\phi(\tau - T_i)} \right]^{N_i}}{(N_i)!} \right],$$

$$f(n) = (\mu\phi)^n \ \exp\left[\phi \sum_{i=1}^{n} (\tau - T_i) - \mu \sum_{i=0}^{n} K_i (1 - e^{-\phi(\tau - T_i)}) \right] \prod_{i=0}^{n} \left[\sum_{j=0}^{i} K_j e^{-\phi(\tau - T_j)} \right]$$

and hence:

$$f(\vec{N} \mid n) = \frac{f(\vec{N},n)}{f(n)}$$

$$= \frac{\phi^n \exp\left[\phi \sum_{i=1}^{n} (\tau - T_i) - \mu \Sigma K_i \right] \prod_{i=0}^{n} \left[\left(\sum_{j=0}^{i-1} N_j - i + 1 \right) \frac{\left[\mu K_i e^{-\phi(\tau - T_i)} \right]^{N_i}}{(N_i)!} \right]}{(\mu\phi)^n \ \exp\left[\phi \sum_{i=1}^{n} (\tau - T_i) - \mu \sum_{i=0}^{n} K_i (1 - e^{-\phi(\tau - T_i)}) \right] \prod_{i=0}^{n} \left[\sum_{j=0}^{i} K_j e^{-\phi(\tau - T_j)} \right]}$$

$$= A^{-1} \prod_{i=0}^{n} \left[\left(\sum_{j=0}^{i-1} N_j - i + 1 \right) \frac{a_i^{N_i}}{(N_i)!} \right], \tag{7.14}$$

where

$$a_i := \mu K_i e^{\phi(\tau - T_i)}, \quad i = 0, 1, \ldots n \tag{7.15}$$

as in (7.5) and

$$A \ := \ \exp\left[\sum_{i=0}^{n} a_i\right] \prod_{i=0}^{n-1}\left[\sum_{j=0}^{i} a_j\right]. \qquad (7.16)$$

With help again of Lemma 7.1 we get from (7.12), (7.14),(7.15) and (7.16):

$$\mathrm{I\!E}\,(N \mid n) \ = \ A^{-1} \sum_{N_0=0}^{\infty} \cdots \sum_{N_n=0}^{\infty} (N_0 + \cdots + N_n) \prod_{i=0}^{n} \left[\Big(\sum_{j=0}^{i-1} N_j - i + 1\Big) \frac{a_i^{N_i}}{(N_i)!} \right]$$

$$= A^{-1} \sum_{N_0=0}^{\infty} N_0 \frac{a_0^{N_0}}{N_0!} \sum_{N_1=0}^{\infty} (N_0 + N_1 - 1) \frac{a_1^{N_1}}{N_1!}$$

$$\times \ \ldots\ldots$$

$$\times \sum_{N_{n-1}=0}^{\infty} (N_0 + N_1 + \cdots + N_{n-1} - (n-1)) \frac{a_{n-1}^{N_{n-1}}}{N_{n-1}!}$$

$$\times \sum_{N_n=0}^{\infty} (N_0 + N_1 + \cdots + N_n - n + n) \frac{a_n^{N_n}}{N_n!}$$

$$= \frac{\exp\left[\sum_{i=0}^{n} a_i\right] \prod_{i=0}^{n}\left[\sum_{j=0}^{i} a_j\right] + n e^{a_n} \exp\left[\sum_{i=0}^{n-1} a_i\right] \prod_{i=0}^{n-1}\left[\sum_{j=0}^{i} a_j\right]}{\exp\left[\sum_{i=0}^{n} a_i\right] \prod_{i=0}^{n-1}\left[\sum_{j=0}^{i} a_j\right]}$$

$$= \sum_{j=0}^{n} a_j \ + \ n$$

and hence:

$$\mathrm{I\!E}\,(N(t-) \mid \mathcal{F}_{t-}) \ = \ \mu \sum_{i=0}^{n(t-)} K_i e^{-\phi(t-T_i)} + n(t-). \qquad (7.17)$$

From (7.11) and (7.17) we now see that the intensity function under the filtration $\mathcal{F}_{t-}$ (only observing the counting process $n(s)$, $0 \le s < t$ and the software changes K_i, $i = 0..n(t-)$) is given by

$$\lambda^{\mathcal{F}}(t) \ := \ \mu\,\phi \sum_{i=0}^{n(t-)} K_i e^{-\phi(t-T_i)}.$$

We note that (7.15) is multiplicative in both μ and the K_i. It seems, however, more realistic, when applying asymptotic theory as developed in Van Pul (1990), to consider the following sequence of increasing experiments $\{(n_\nu(t),\lambda_\nu(t)),t\in[0,\tau]\}$, $\nu=1,2,\ldots$. We define the intensity function of the ν-th experiment by

$$\lambda_\nu^{\mathcal{T}}(t) := \mu_\nu\ \phi \sum_{i=0}^{n(t-)} K_{i,\nu}\mathrm{e}^{-\phi(t-T_i)}, \tag{7.18}$$

where

$$\mu_\nu := \nu^{-\alpha}\mu \tag{7.19}$$

$$K_{0,\nu} := \nu^{1+\alpha}\,K_0, \tag{7.20}$$

$$K_{i,\nu} := \nu^{\alpha}\,K_i,\quad i=1,\ldots n_\nu(\tau), \tag{7.21}$$

for some $\alpha\geq 0$, $\nu=1,2,\ldots$. That is, we consider a sequence of experiments, where $K_{0,\nu}$, the initial software size in the ν-th experiment, is increasing faster than the other $K_{i,\nu}, i=1,2,\ldots$ in such a way that - analogous to the choice of asymptotics in Van Pul (1990) - the expectation of $N_{0,\nu}$ tends to infinity, but $\mathbb{E}(N_{i,\nu})=\mu K_i$ remains constant as $\nu\to\infty$. This seems reasonable because $K_0\gg K_i$ in practice. Defining

$$k_\nu(t) := \sum_{i=1}^{n_\nu(\tau)} K_{i,\nu}(t=T_i),\quad t\in[0,\tau],$$

$$x_\nu(t) := \nu^{-1}n_\nu(t),\quad t\in[0,\tau],$$

we find that the intensity function of the PG&IR model

$$\lambda_\nu(t) := \nu\mu\phi K_0\mathrm{e}^{-\phi t} + \mu\phi\sum_{i=1}^{n_\nu(t-)} K_{i,\nu}\mathrm{e}^{-\phi(t-T_i)}$$

$$= \nu\mu\phi\mathrm{e}^{-\phi t}\left[K_0+\int_0^{t-} k(s)\mathrm{e}^{\phi s}dx_\nu(s)\right]$$

is indeed of the special form (7.10) with

$$\beta(t;\mu,\phi;x(t-)) := \mu\phi\mathrm{e}^{-\phi t}\left[K_0+\int_0^{t-} k(s)\mathrm{e}^{\phi s}dx(s)\right], \tag{7.22}$$

independent of the choice for α. We can now formulate the main result of this paper.

Theorem 7.1 *Let $\tau > 0$. We assume that the failure data are generated by the PG&IR intensity function β in (7.22) with true parameter value $\theta_0 := (\mu_0, \phi_0)$, satisfying $0 < \mu_0 < \infty$ and $0 < \phi_0 < \infty$. Then the ML-estimators $\hat{\mu}$ and $\hat{\phi}$ suggested by the last two equations of Section 7.3, are consistent, asymptotically normal distributed and efficient.*

Proof Theorem 7.1 The proof of Theorem 7.1 is tedious but routine and therefore omitted here. With the following choice for Θ_0:

$$\Theta_0 := [\varepsilon_\mu, M_\mu] \times [\varepsilon_\phi, M_\phi], \quad 0<\varepsilon_\mu<\mu_0<M_\mu<\infty, \quad 0<\varepsilon_\phi<\phi_0<M_\phi<1,$$

the desired result follows immediately from Van Pul (1990) by verifying the conditions (GC1)-(GC2)&(LC1)-(LC3) for the intensity function β in (7.22) and applying Theorem □

Let us consider the PG&IR model again as given by (7.1)-(7.3). Note that the process $N(t)$ is unobservable. Thus defining the filtrations

$$\mathcal{F}_{t-} := \{ n(s) : 0 \le s < t \},$$

$$\mathcal{G}_{t-} := \{ n(s), N(s) : 0 \le s < t \},$$

we notice that the intensity λ given in (7.3) is actually $\lambda^{\mathcal{G}}$, the intensity function of the counting process with respect to the filtration $\mathcal{G}_t$. With use of the Innovation Theorem (see e.g. Bremaud (1977)), and another application of Lemma 7.1 we can show that the intensity function under the filtration $\mathcal{F}_{t-}$ (only observing the counting process $n(s)$, $0 \le s < t$ and the software changes K_i, $i = 0..n(t-)$) is given by

$$\lambda^{\mathcal{F}}(t) := \mu\,\phi \sum_{i=0}^{n(t-)} K_i e^{-\phi(t-T_i)}.$$

An interesting idea seems to set all the K_i equal to some $\bar{K}$ except for $K_0 \gg \bar{K}$. With parameters $N_0 := \mu K_0$ and $\bar{N} := \mu\bar{K}$ the failure intensity becomes

$$\lambda(t;\phi,N_0,\bar{N}) := N_0 \phi e^{-\phi t} + \bar{N}\phi \sum_{i=1}^{n(t-)} K_i e^{-\phi(t-T_i)}.$$

In this three parameter model, $\bar{N}$, the average number of faults introduced per repair action, can be interpreted to account for dependencies between faults. Whenever hidden faults become observable because of a fault repair, this can be considered as the introduction of new faults. Finally note that for $\bar{N}=0$ the above model reduces to the well-known model of Goel-Okumoto (1979).

7.5 Regression models

The PG&IR model can be seen as a a special case within a general class of regression models. In the previous section we assumed that the N_i were Poisson distributed with a parameter depending on a single software measure. Because the process of introducing new faults is so difficult to understand, it seems appropriate to use explanatory variables and apply regression analysis. We therefore suggest the following class of models given by (7.1)-(7.3) and

$$\mathbf{N_i} =_d POI(X_i),$$

$$X_i := \exp\left[\beta_1 z_i^1 + \ldots + \beta_m z_i^m\right],$$

where the $z_i^j, j=1 \ldots m$, are the known realisations of m software measures Z^j (like e.g. size, complexity, number-of-loops) at time T_i and where the $\beta_j, j=1 \ldots m$, denote the corresponding regression coefficients we have to estimate. Statistical methods are available to investigate whether certain explanatory variables are redundant (or not) and whether their influence is linear, via another power, or say logarithmic.

7.6 Open problems and concluding remarks

We have constructed a model, which is able do deal with imperfect repair and software growth. Moreover, we have shown that the ML-estimators for the model parameters have desirable asymptotic properties. Immediately a number of questions arise:

(1) How fast will the asymptotic behaviour appear in practice? Already for a very simple model as the Musa model the convergence to the asymptotic normal situation can be pretty slow (see Van Pul (1991a)). Can we construct confidence intervals with satisfying coverage percentages ?

(2) Our asymptotic approach seemed natural, but of course there are other ways to deal with it. By our choices in (7.18)-(7.21) the model becomes asymptotically equivalent with the well-known model of Goel & Okumoto (1979) and we have thus lost a lot of information. Is it possible to bring the intensity function in the required form (7.10) without losing the contribution of the K_i, $i=1,2,\ldots$?

(3) Is it worthwhile at all to search for sophisticated models incorporating imperfect repair and software growth, when estimation of parameters even in simple models can be so disappointing ?

Trying to construct a more or less realistic model for the reliability evolution during testing with attractive (asymptotic) statistical properties seems to be a good aim. Getting nice simulation results or good behaviour in practice, however, does apparently not immediately follow from this.

In the field of regression models for software reliability, there is in my opinion a lot of interesting research still to be done. Essential will be, however, the collection of real data (computation of various software measures) by software developers. So far, we did not get much response from them. Perhaps they should read Rook's (1990) Handbook on Software Reliability. In its preface Boehm resignedly states: *"Sometime soon, software reliability is going to become a highly visible and important field. Unfortunately, given human nature, its thrust into prominence will only happen once we experience the software equivalent of the Chernobyl, Bhopal, or space shuttle Challenger disasters. Such a disaster is likely to happen in the next few years..."*.

Part III

Practical Application

Chapter 8

Two Case-Studies

8.1 Introduction

Lastly we describe two case-studies performed at companies in The Netherlands which are developing large computer systems: Philips Medical Systems (PMS) and Ericsson Telecommunications (ETM). We investigate whether the models and statistical methods, developed earlier (Van Pul (1992ab)), can be applied or not, and we eventually give recommendations how the software development and software test processes should be adapted to get better results.

This chapter is organised as follows. In the next section we introduce some mathematics, concerning how to deal with *grouped data* as is the case in the two case-studies. In Sections 8.3 and 8.4 we describe the case-studies at PMS and ETM, respectively. Finally, in the last Section 8.5 we sketch some of the practical problems encountered, we give some recommendations and concluding remarks.

8.2 Some mathematical background on counting processes with grouped data

Using the test-data available we will compare the Jelinski-Moranda (JM) and Littlewood (L) model. Briefly rephrasing, the failure rate and mean value function for the Jelinski-Moranda model are given by

$$\lambda^{\mathrm{JM}}(t) = \phi \left[N - n(t-) \right] ,$$

$$\mathbb{E}\, n(t) \ = \ N \left[1 - \exp(-\phi t) \right];$$

whereas in the Littlewood model these functions can be expressed as

$$\lambda^{\mathrm{L}}(t) \ = \ \frac{\alpha}{1 + \varepsilon\, t} \left[N - n(t-) \right],$$

$$\mathbb{E}\, n(t) \ = \ N \left[1 - (1 + \varepsilon\, t)^{-\alpha/\varepsilon} \right].$$

For a more detailed discussion of these models and of the main software reliability concepts we refer to Musa et al. (1987) or to Chapter 2 of this thesis. We will compute estimators for the model parameters using the method of Maximum Likelihood Estimation. From the test-data, as we will soon see, it follows that we will encounter two significant deviations from classical theory. First of all, we do not observe exact failure times but only the number of the day on which a failure occurs. Secondly, the test intensity is not constant during the testing period. To be more formal: For each of K working-days, we are given n_i, the cumulative number of software faults experienced up to and including day i, and b_i, the cumulative number of test blocks performed up to and including day i.

As the data are essentially grouped, expressions for the classical likelihood seem, at first, not to be of use. Binomial-type models (such as the Jelinski-Moranda model (1972) and Littlewood model (1980)) assume that the number of failures experienced by time t follows a binomial distribution. This assumption yields the following expression for the likelihood function for grouped data:

$$L(N,\theta) \ := \ \prod_{j=1}^{K} \binom{N-n_{j-1}}{n_j-n_{j-1}} \ [F_\theta(b_j \,|\, b_{j-1})]^{n_j-n_{j-1}} \ [1-F_\theta(b_j \,|\, b_{j-1})]^{N-n_j}, \qquad (8.1)$$

where

$$F_\theta(t\,|\,s) \ := \ \frac{F_\theta(t)-F_\theta(s)}{1-F_\theta(s)}. \qquad (8.2)$$

In (8.2) $F_\theta(t)$ denotes the probability distribution of the failure times, and is hence model dependent. For the Jelinski-Moranda model (JM) and the Littlewood model (LW) these are respectively the exponential and generalised pareto distribution:

$$F^{JM}(t) \ := \ 1-\mathrm{e}^{-\phi t}, \quad \theta=\phi;$$

$$F^{LW}(t) \ := \ 1-\left(\frac{1}{1+\varepsilon t} \right)^{\alpha/\varepsilon}, \quad \theta=(\alpha,\varepsilon).$$

Finding maximum likelihood estimators is usually done by taking the log of (8.1) and setting partial derivatives equal to zero. In case we assume constant test intensity, that is $b_j=\alpha j$ for some proportional constant α, the system of log-likelihood equations becomes numerically more tractable as a lot of dependence on j is eliminated.

We will, however, follow another approach. Suppose we have found n_j-n_{j-1} bugs in test interval (b_{j-1},b_j). We will consider the counting process with exactly n_j-n_{j-1} jumps of size 1 at the centres of the test interval $(b_{j-1}+b_j)/2$, or at equidistant points $b_{j-1}+k(b_j-b_{j-1})/(n_j-n_{j-1})$, $k=1,...,n_j-n_{j-1}$. That is, we act as if individual data represent exact failure times T_i and apply the conventional formulas for the likelihood (see Van Pul (1992a)). As the number of groups K (=50) is quite large, we expect that differences in results of the approaches described will be small, and in practical situations negligible with respect to deviations caused by erroneous and incomplete data. However, as we suspect that the test intensity data (e.g. the $b_j, j=1,...,K$) are not very reliable we will also compare the different approaches (i) assuming the test intensity is constant and (ii) estimating the test intensity by the number of test blocks.

8.3 A software reliability case-study at Philips Medical Systems (PMS)

Nowadays software is used in all kinds of systems, which improve the quality of life. A good example of this is the increasing use of computer systems in medical imaging applications such as X-ray, MRI and ultrasound scanners. It is obvious that the software involved should be highly reliable. In this case-study we will investigate the failure behaviour of the application software for a particular system development at Philips Medical Systems (PMS) in Best (The Netherlands), during a particular phase of the testing process. In Section 8.3.1 we briefly describe how software development and testing is organised at PMS and which test data were collected. In Section 8.3.2 we will present most of the collected data in various forms, tables and figures, whereas in Section 8.3.3 estimations and predictions are given of some characteristics of the software.

8.3.1 Software development and testing at PMS

PMS has started the development of products containing large software packages in the late seventies. Of these products about every year an updated version is released. Each release supports new hardware, introduces new functionality, but also includes internal structural improvements. From annual project-start up to release one can roughly distinguish the following development and test phases:

(a) *development phase 1* (several months)
(b) *pi-test* (several weeks): Preliminary installation test. This is a test to get a basic part of the system operational and ready for a clinical tryout on some probe sites.
(c) *development phase 2* (several months)
(d) *alpha-test* (several weeks): This is a software test performed by the software developers. Hardware is not tested, but only used to test the software.
(e) *SIT* (a few weeks): System integration test. Software is run on the definitive hardware with the aim of testing the complete system.

(f) *beta-test* (a few weeks): During the beta-test the complete system is tested by the software management group, which is responsible for released software products, by service and manufacturing in co-operation with people from outside (application specialists and system-operators of the instrumentation in hospitals). This test serves mostly as acceptance test. Hereafter the new product is released.

In practice there is some limited, controlled overlap between development- and testing phases. Sometimes the alpha-test and the SIT are combined. In this case-study we will consider the failure behaviour of the application software (ASW) during the pi-test and alpha-test; these are the phases where most of the software failures are detected and corrected. The application software (ASW) is divide into several subsegments. See Table 8.1. An eloc stands for an executable line of code. Note that during development a large proportion of the elocs is left unchanged.

subsegment	# modules	# kilo elocs
1	ca. 330	ca. 60
2	ca. 300	ca. 50
3	ca. 130	ca. 50
4	ca. 180	ca. 40
5	ca. 180	ca. 20
6	ca. 90	ca. 20
7	ca. 80	ca. 20
8	ca. 30	ca. 10
9	ca. 20	ca. 10
10	ca. 10	ca. 10

Table 8.1
Number of modules and kilo elocs for some subsegments of ASW.

Before alpha-testing starts, the test-leader has made a thorough alpha-test-plan. This plan states exactly which subsegment is tested by whom and when. Time is also reserved for problem solving. The software developers are assumed to prepare the test sessions for the testers by writing tests specs. Software bugs are reported via an (Internal) Problem Report ((I)PR) and gathered in a database. Moreover, testers have to fill in a detailed log-form with information such as times of start and end of test, description of test items, which of them were successful and which not, and eventually other comments concerning (mal) functioning of hardware. Twice a week newly entered PR's are inspected at a so called Software Progress Meeting and assigned to the for this bug most capable problem solver.

Both the pi-test and alpha-test under consideration lasted about 10 weeks (50 working days). During these test-phases for each occurring software bug the following test data were collected in the PR database:

(1) The problem report number that identifies the fault.
(2) The occurrence date of the problem. Note that usually in software reliability failure times are gathered. Here we are confronted with "grouped data", the total number of

faults detected per day. We will discuss the mathematical consequences of this in the next section.

(3) The subsegment to which the PR is assigned.

(4) The priority of the fault: routine (RO), urgent (UR) or very urgent (VU). Stack-dumps are typically very urgent; minor display problems routine. The priority of faults may be changed as deadlines come closer. The priority field is hence used to control SW engineers and therefore is not a very objective measure.

Moreover, during the alpha-test via the log-forms record was kept of

(5) The total test effort per day in testing hours.

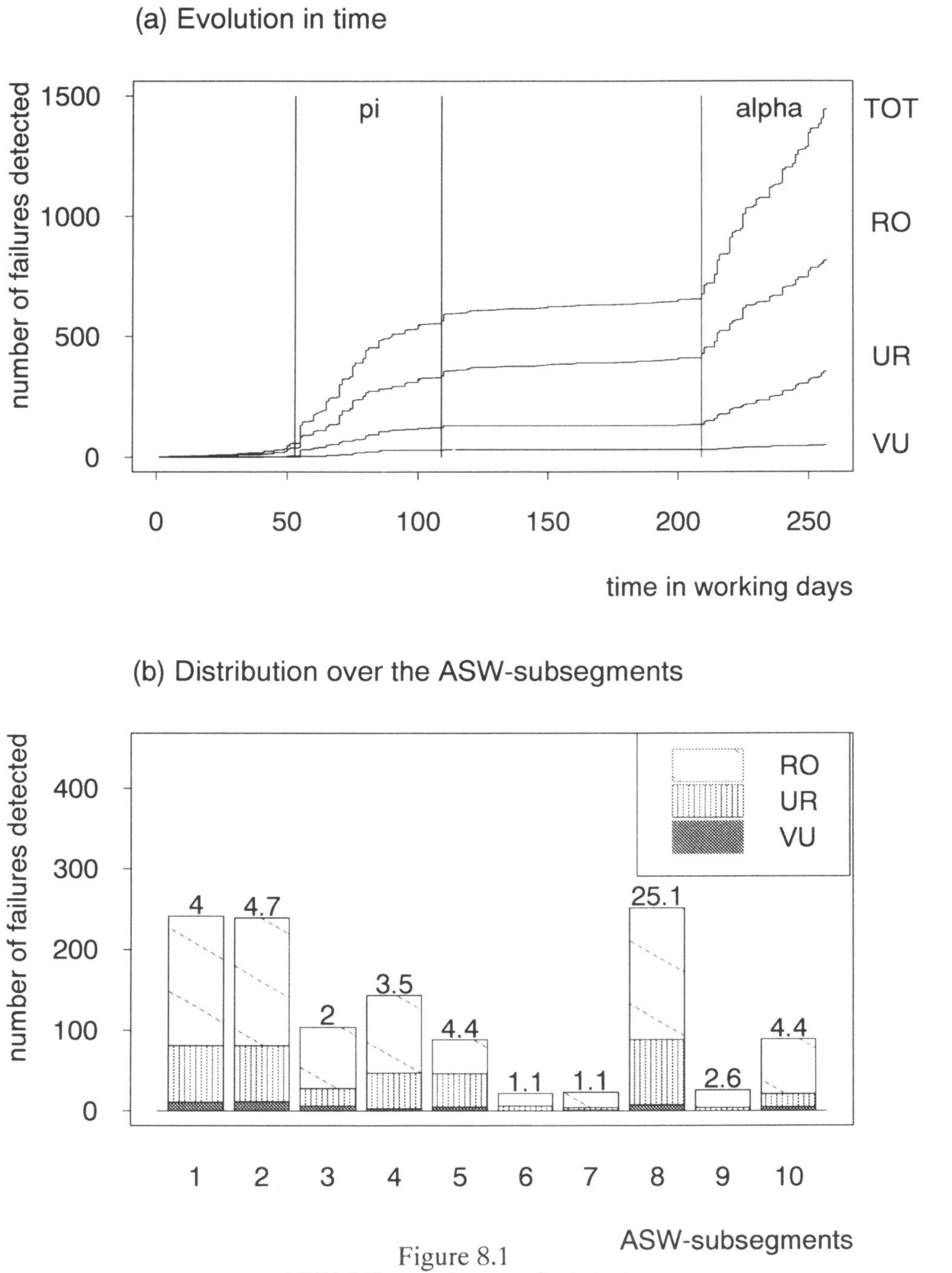
We had in mind to collect also other test data such as the solving time of a problem (in man-hours), the size of the change in the software (in lines of code) and an estimation of the occurrence probability of each fault. The collection of these data caused either too much practical difficulties or caused that the significance of the data would be highly doubtful, due to inevitable subjectivities. It should be mentioned that also in the data collected some subjectivity (or non-consistency) could not be excluded completely. Moreover, the data of (5) suffer from inaccuracy, due to late or incomplete submission of the log-forms. We will return to this later, when discussing the results.

8.3.2 The test data

Before concentrating ourselves completely on the data collected during the pi- and alpha-test of the current release, we will give a global idea of the failure history from project start. After about half a year of development a first test phase, the so-called pi-test, was performed for several weeks. After another development period of several months, a second test phase, the alpha-test began. The complete failure-history from project start is visualised in Figure 8.1a. The first bug was found and reported near the end of development phase 1. The upper dotted line represents the total number of bugs found versus time; the other three lines correspond to the number of bugs with priority routine (RO), urgent (UR) and very urgent (VU). Time is giving in working days, that is weekends and holidays are not counted.

ASW is divided in a number of subsegments. In Figure 8.1b the distribution of the ASW-bugs over those subsegments is shown. The figures above the bars represent number of faults per 1000 elocs of existing code. The relatively large number 25.1 for subsegment 8 (in comparison with other subsegments), indicating a high failure intensity, may be explained by the fact that subsegment 8 consists of almost 100% new software and only little reuse of old software. See Grady & Caswell (1987) for generally accepted figures of statistical prediction models on productivity and reliability.

In Figure 8.2 the cumulative number of faults detected is plotted against the time in working days. The failure rate seems to decrease during pi-test, but remains constant during alpha-test. This can be explained by the fact that during the alpha-test the functionality of the system is tested in a very systematic way, part by part. Hence every now and then a new subsegment is inspected. Therefore the number of detected failures will tend to grow linearly in time.



(a) Evolution in time

(b) Distribution over the ASW-subsegments

Figure 8.1
ASW-failures since project start
(a) Evolution in time (b) Distribution over the ASW-subsegments.

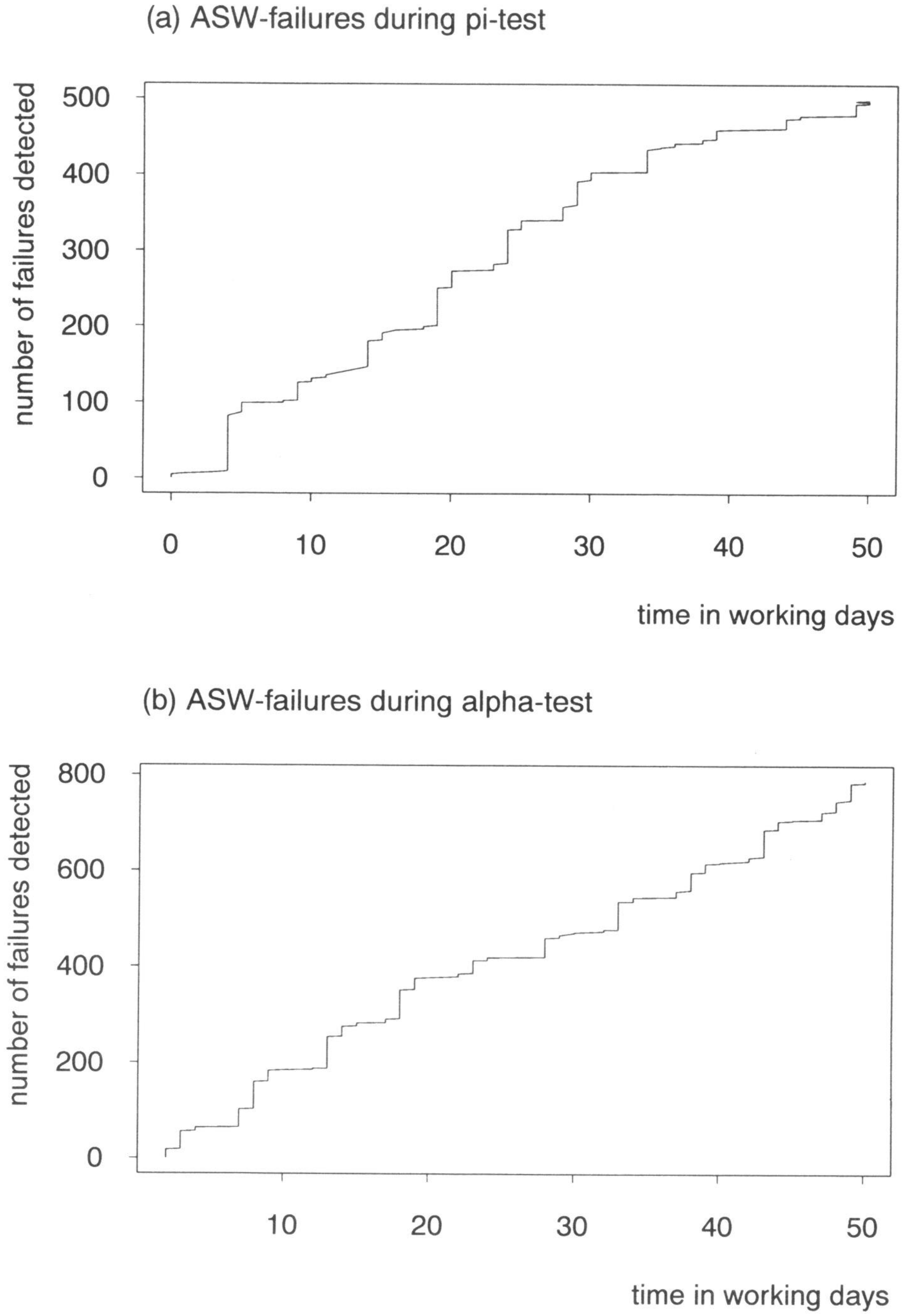


Figure 8.2
Number of ASW-failures detected during
(a) pi-test and (b) alpha-test.

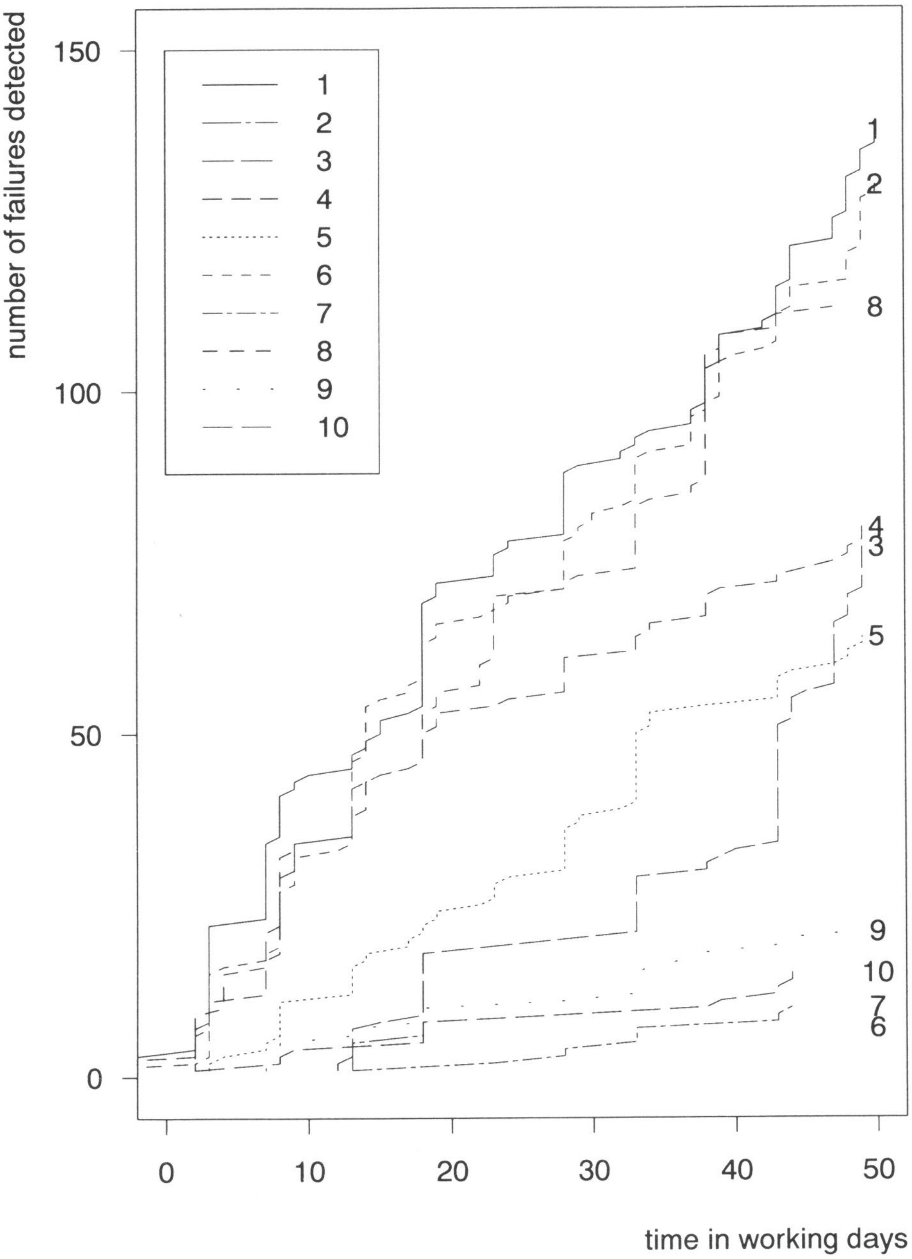


Figure 8.3
ASW-failures during alpha test.
Evolution in time for the different ASW subsegments.

(a)	RO	UR	VU	TOTAL
1	62	20	4	86
2	65	23	6	94
3	13	3	4	20
4	33	15	2	50
5	12	3	0	15
6	12	1	0	13
7	13	0	0	13
8	73	47	7	127
9	4	1	0	5
10	40	12	5	57
TOTAL	327	125	28	480
(b)	RO	UR	VU	TOTAL
1	82	46	7	135
2	80	46	4	130
3	59	16	2	77
4	47	29	1	77
5	25	37	2	64
6	2	5	0	7
7	6	3	1	10
8	78	32	1	111
9	18	3	0	21
10	12	3	0	15
TOTAL	409	220	18	647

Table 8.2
Classification of ASW-failures during
(a) pi-test (b) alpha-test.

In Figure 8.3 the failure behaviour of individual subsegments during alpha test is compared. The three most critical subsegments 1, 2 and 8, have almost identical failure behaviour. The same can be said about the four smallest subsegments, 6, 7, 9 and 10. The three remaining subsegments, 3, 4 and 5, subsegments of about same size and number of bugs found, show however different failure histories. The failure intensity of subsegment 5 seems to be constant (as seems the case for most of the subsegments in this project), the failure intensity of subsegment 4 is nicely decreasing, whereas the failure intensity of subsegment 3 is still growing. This suggests, that at least for this subsegment, the alpha test was too short. Indeed, in sequel tests (SIT and beta) relatively large numbers of component 3 faults were observed.

In Figure 8.4 we compare two different time scales: time in working days and time in testing hours. We see that the test intensity during alpha-test in the first three and last three weeks was significantly larger, due to hardware problems in the weeks between. The test intensity grew gradually from, say day 15. Cumulative faults seem therefore to grow linearly with time (in working days), but negative exponentially with cumulative test effort (in testing hours). In Section 8.3.3 we will investigate whether these two approaches (working days versus testing hours) for the ASW data during alpha-test, will lead to significant differences in the statistical estimations and predictions.

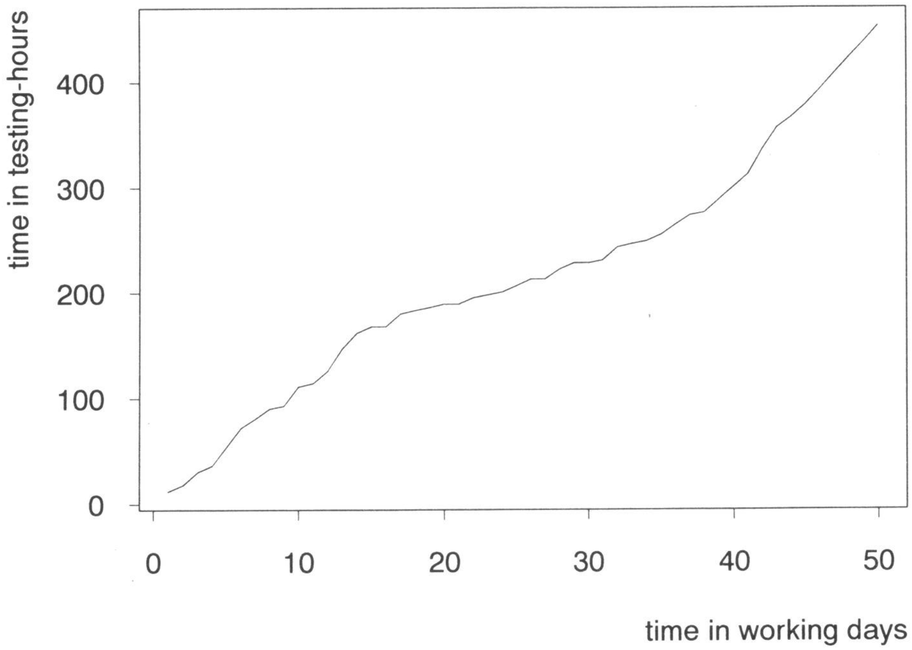


Figure 8.4
Time in working days versus time in test hours.

8.3.3 Statistical inference

In this section we will compare the Jelinski-Moranda (JM) and the Littlewood model (L), mentioned earlier. We will try to fit both models to the PMS data of pi- and alpha-test. As during alpha-test also testing-hours were registered we will investigate the following two approaches:

(i) Assume the test intensity is constant (time in working days).

(ii) Estimate the test intensity by the number of testing hours.

Estimation of the model parameters is -as usually- done by the method of Maximum Likelihood Estimation (MLE). For background on this method and computation of the likelihood functions of the two models, we refer to Van Pul (1992a, 1992b). Parameter estimations and standard deviations are given in Table 8.3a and 8.3b.

JM	pi	alpha (i)	alpha (ii)
$\hat{N}$	533 ± 27	3480±2082	1646±324
$\hat{\phi}$	0.0316±0.0035	0.0042±0.0028	0.0012±0.0003
$\hat{\lambda}(\tau)$	3.35	11.89	1.16

L	pi	alpha (i)	alpha (ii)
$\hat{N}$	525±26	2381±887	1454±235
$\hat{\alpha}$	0.0226±0.0024	0.0064±0.0028	0.0014±0.0003
$\hat{\varepsilon}$	0	0	0
$\hat{\lambda}(\tau)$	3.20	11.12	1.09

Table 8.3
Parameter estimations and standard deviations for the model of Jelinski-Moranda (JM) and Littlewood (L).

From Table 8.3 we see that when trying to fit the Littlewood model, the parameter ε turns out to be zero, indicating that the Jelinski-Moranda model is the best model within the larger Littlewood class. The differences in the estimates of N and ϕ between Table 8.3a and 8.3b come from instabilities of the numerical procedures involved and their stop-criteria. Contour curves of the likelihood function in the neighbourhood of the maximum take the form of extremely flat ellipses. The estimation of the failure intensity λ, a software characteristic which seems more relevant than just the total number of bugs N, is more stable. Covariances for the model parameters can be easily derived. See again Van Pul (1992b). Although we were suspicious about the accuracy of the test-intensity data, it turns out out that using approach (ii) reduces both estimate of N and its variance. Standard deviations for $\hat{N}$ of the pi-test are very reasonable; those of the alpha-test quite large, showing again that the alpha-test was not the most appropriate test phase to apply software reliability theory.

In Figure 8.5 we give estimates and 95% confidence bands of the mean value function for the pi-test data. In Figure 8.6 and 8.7 we do the same for the alpha-test data, using approach (i) and (ii) respectively. Further research would be needed to develop a prediction band for the future course of the stochastic process n itself. From mathematical point of view the pi-test seems to be very appropriate to predict future software behaviour. In practice, however, immediately after the pi-test a second large development phase takes place. Figure 8.7 suggests that another test effort of 500 or 1000 testing hours would yield about 400 or 700 new bugs, respectively. In Figure 8.6 also the total number of bugs detected at the end of SIT and beta-test are indicated: respectively 1068 and 1523. We see that these added points are not far from the predicted values. As exact figures for the test intensity during SIT and beta-test, were not available, we could not add these two points to Figure 8.7.

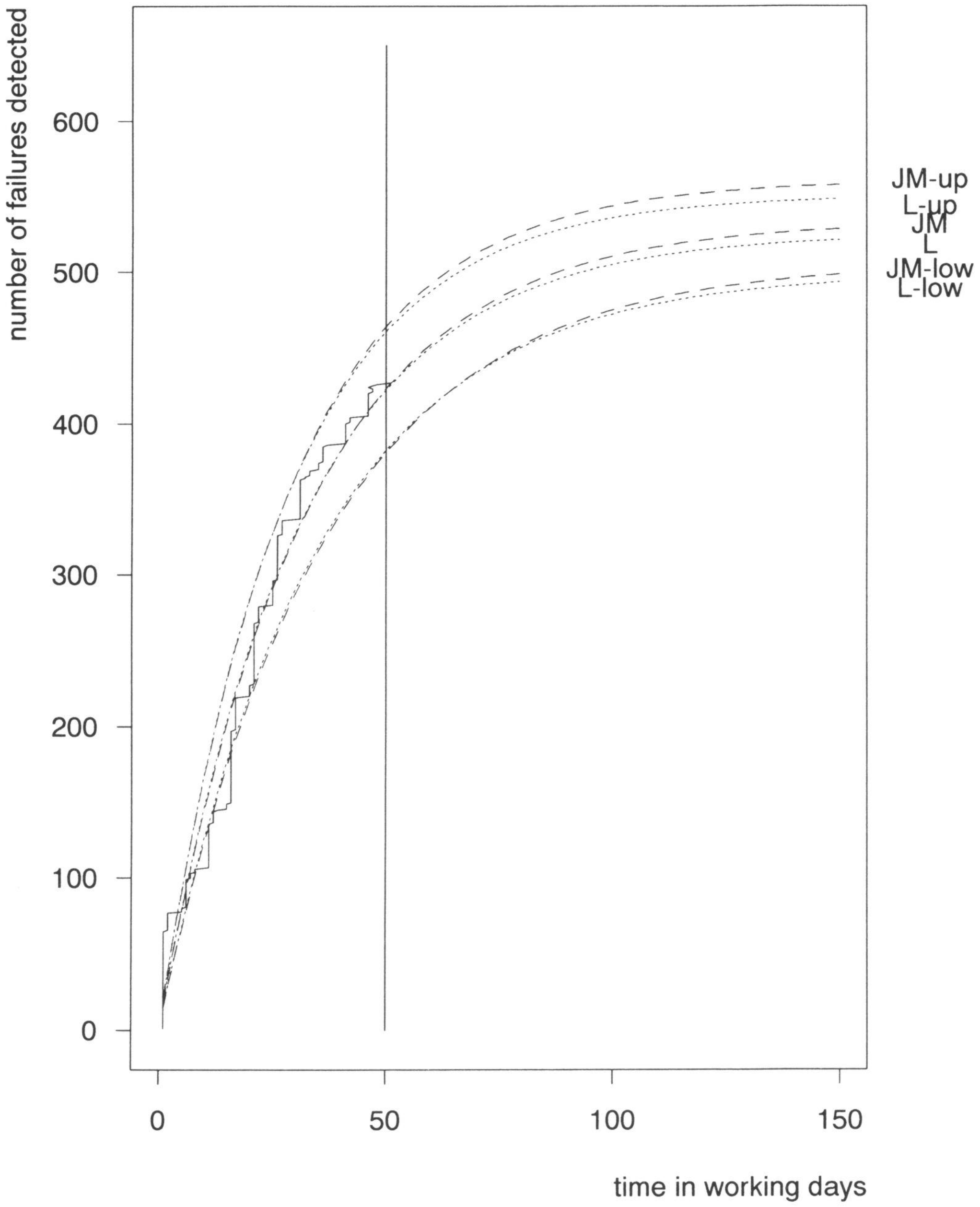


Figure 8.5
Estimates and 95% confidence bands for the mean value function of the Jelinski-Moranda and Littlewood models for the pi-test data.

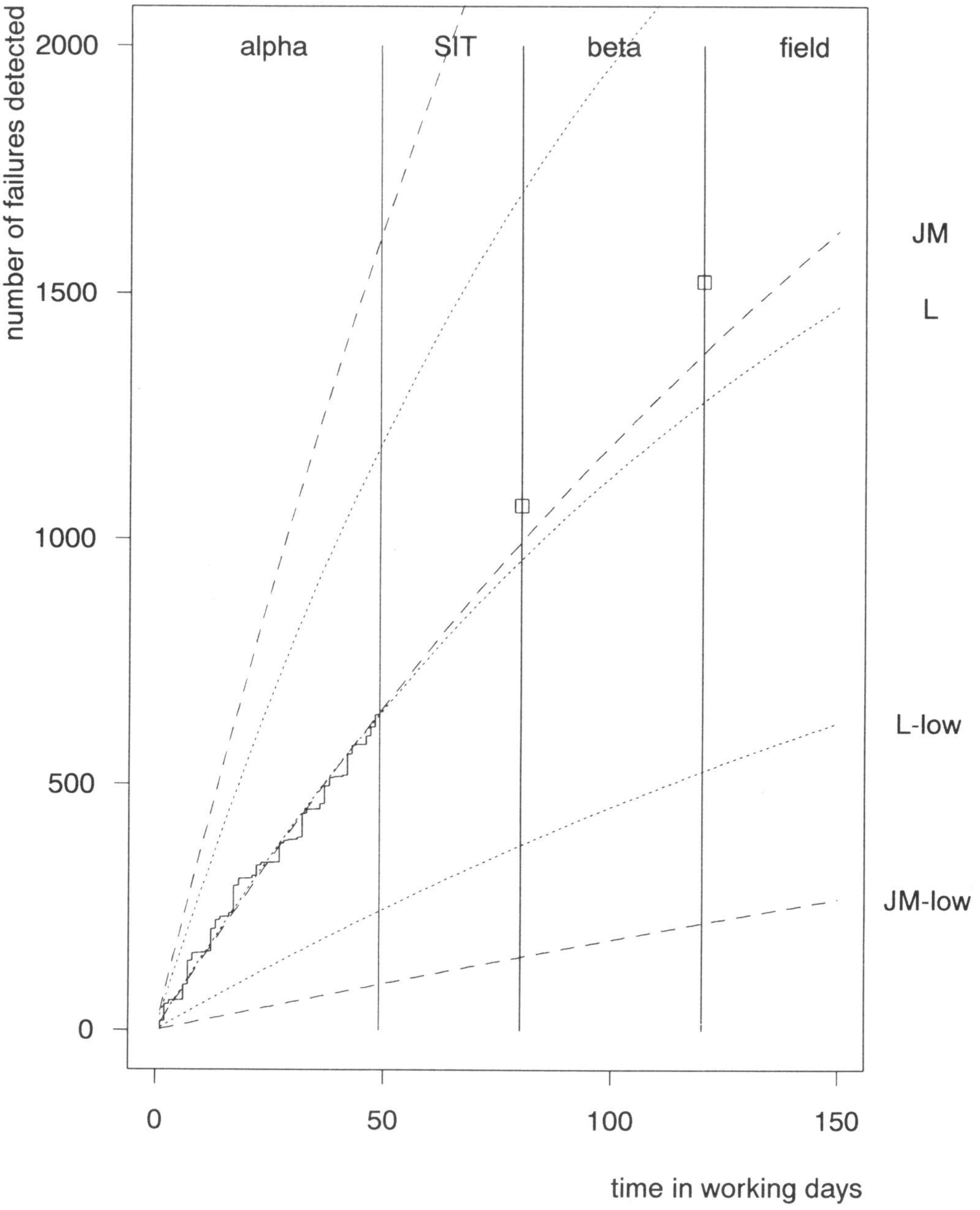


Figure 8.6
Estimates and 95% confidence bands for the mean value function of the Jelinski-Moranda and Littlewood models for the alpha-test data using approach (i).

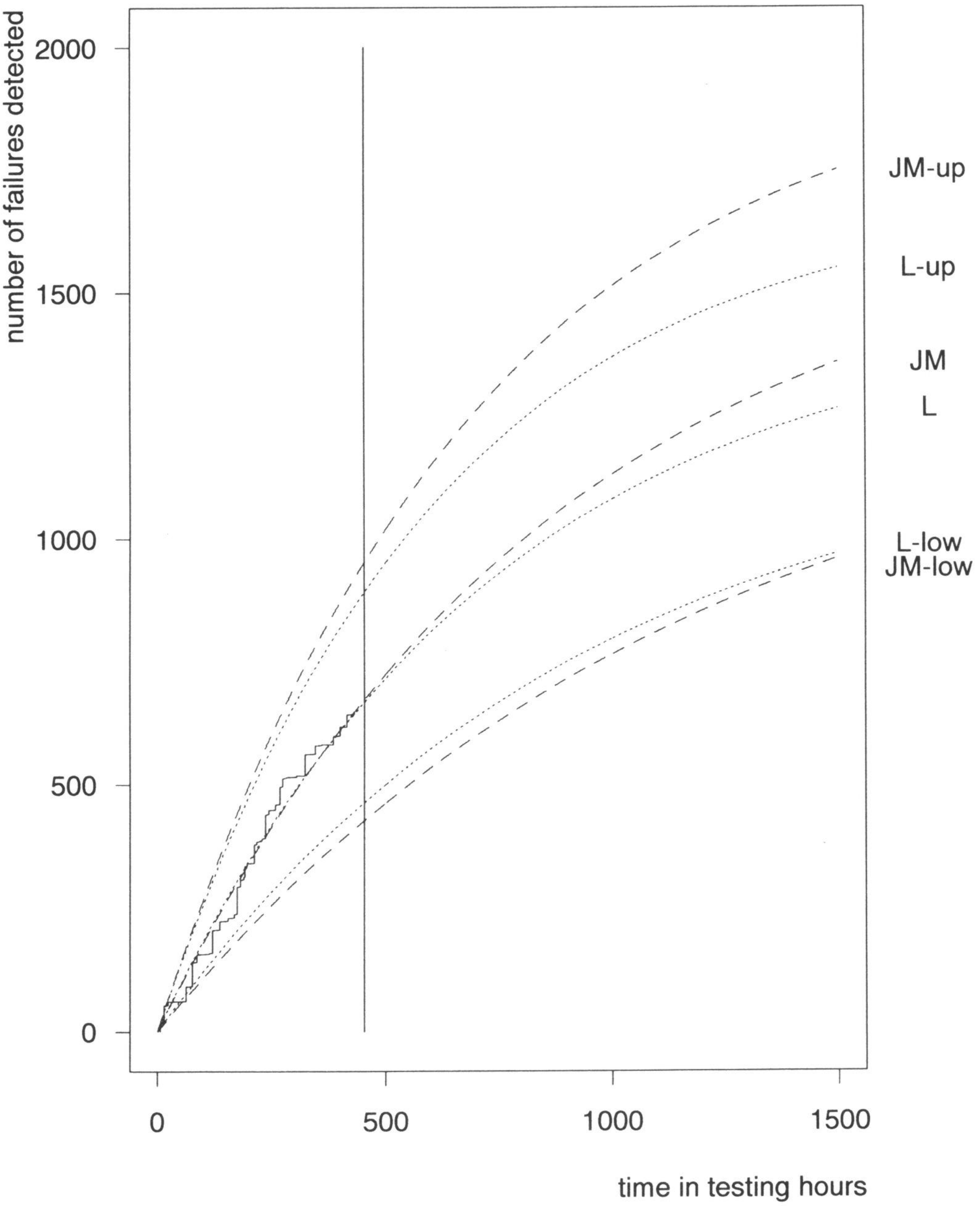


Figure 8.7
Estimates and 95% confidence bands for the mean value function of the Jelinski-Moranda and Littlewood models for the alpha-test data using approach (ii).

8.4 A Software reliability case-study at Ericsson Telecommunications (ETM)

In today's world both industrial and social concerns often go far beyond the seeable horizon. Therefore telecommunications, which can be defined as the exchange of information over a distance using electrical means, has become a very important field in modern technology. Satellites, optical fibre and digital exchanges are just a few examples indicating that this field is still evolving very fast. Of course, as in any other field nowadays, computer systems have taken over most human tasks.

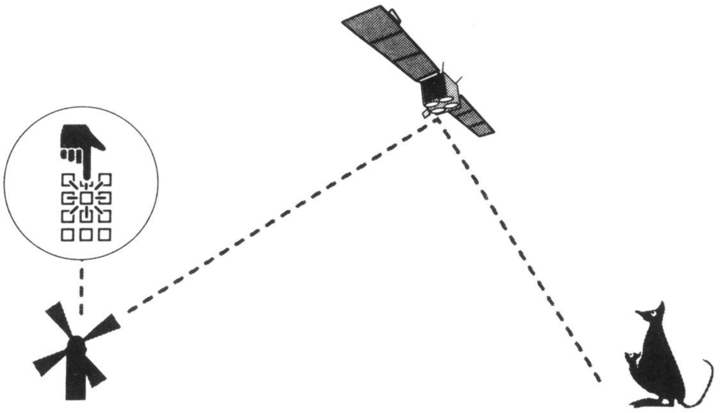


Figure 8.8
Telecommunications today.

There are various sources for temporary problems in telecommunications. At rare occasions (such as January 1st, after an earthquake or during certain popular TV-shows) many subscribers want to communicate at the same time (overload). Sometimes the transmission equipment is damaged by construction activities or bad weather. However, the problems are mostly caused by hardware or software faults in the exchange. Much of the hardware in modern exchanges is duplicated so that if a hardware fault occurs, standby units can be put into service in place of those which are faulty. Hardware faults can thus be repaired without interrupting service. Software errors, which typically occur shortly after a new system is put into operation, are more serious and can cause the exchange to be out of action for several hours.

In this case-study we will investigate the failure behaviour of the software in a new application system developed for a particular telephone exchange system (AXE) at Ericsson Telecommunications (ETM) in Rijen (The Netherlands), during the final stage of its testing process. In Section 8.4.1 we will briefly describe how the development and test process at ETM are organised and which test-data were collected. We will present most of the data collected in Section 8.4.2 and in Section 8.4.3 estimations and predictions are given of some characteristics of the software.

8.4.1 Software development and testing at ETM

In the second part of the nineteenth century the telephone and the Swedish company Ericsson were born almost at the same time. Since then, Ericsson has developed telephones, transmission systems and exchanges. As speech is analogue at its source, telecommunication transmission techniques have been also for a long time. Digital transmission media, however, can carry more traffic and at a higher speed. Besides, digital transmission systems are more reliable than analogue systems because they are less affected by noise and attenuation. Noise is unwanted electrical energy which is present when a signal is being carried on a medium; attenuation is the loss of power (or fading) which a signal experiences during transmission. In 1977 the first Dutch digital Exchange was delivered, the AXE.

AXE is actually the family name for a variety of exchanges with only their name and architecture in common. There are special AXE variants for car telephony, international, intercontinental and 06 numbers (free-phone, tariff-phone, etcetera). Special hardware and software enables them to cope with the different protocols of foreign exchanges or with a variety of tariff structures. A goal of the telecommunications industry is to integrate the existing services (as telephone, telex, fax, etcetera) into one network. One of the most recent additions in telecommunications is the Integrated Services Digital Network (ISDN), which provides such a suitable interface for transmitting speech, data, images and text. See Figure 8.9.

The product under consideration in this case-study is a new application system for a digital exchange which fits in the concept of ISDN. An Application System (AS) is a collection of products designed to meet all the requirements of a certain market. This collection serves as a basis for the creation of a number of functionally similar exchanges.

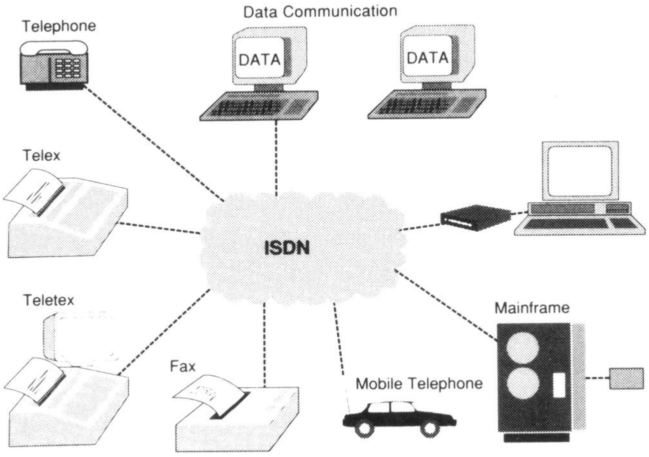


Figure 8.9
ISDN.

At ETM, software development and testing takes place according to a hierarchical decomposition-integration strategy. We will explain what we mean by that. An AXE application system (system level 1) consists of two main parts: switching equipment for switching telephone calls (APT) and a computer for controlling the switching equipment (APZ). Both APT and APZ (systems level 2) use hardware (printed board assemblies) and software (programs and data) and are divided into a number of subsystems, see Table 8.4. An eloc stands for an executable line of code.

Subsystem	# blocks	# kilo elocs
OMS	83	114
SSS	62	79
TSS	40	59
CCS	37	44
GSS	20	52
CHS	20	52
TCS	18	24
NMS	15	18

Table 8.4
Number of blocks and kilo elocs for some subsystems of the application system.

Each subsystem is in turn divided into a number of parts called function blocks. Finally, function blocks consist of one or more function units. If the problem is specified and implemented to the lowest level of the hierarchical structure (namely that of the function unit), so-called function unit tests are performed. Hereafter function units are integrated to function blocks. Similarly, after the function block tests, function blocks are integrated to subsystems and so on. The final test before the product gets into customers' hands is the application system verification (ASV). The aim of the ASV is hence to ensure high quality of the functional contents of a whole application system before delivery to the first exchange. The verification concentrates on test of functions and characteristics in the application system, based on functional specifications and market requirements. For this case-study we collected the test-data of this seven week lasting ASV.

Software failures during ASV are reported via Trouble Reports (TR) and gathered in a database. Trouble report entries are among others: date of occurrence, priority and location of the fault. As the total number of software faults during ASV is limited, we will not consider the code indicating which subsystem/function block/function unit caused the failure, but only use date of occurrence and priority for our investigations. Furthermore, the test leader kept a logbook of special events, such as serious hardware problems or unexpected changes in the specification, which obviously would influence the test-process. Finally also the total number of working hours per week was registered.

8.4.2 The test-data

Test data were gathered during seven weeks of ASV. Due to hardware problems or urgent corrections, only 23 days in this period were effectively used for testing. From the total number of working hours per week We estimated the total number of testing hours per week by multiplying the total number of working hours per week by the fraction of days which were effectively used for testing in that particular week. Occurrence dates and priorities of faults were stored, see Table 8.5.

At Ericsson three priority-classes are distinguished, so-called A, B and C faults, representing major, intermediate and minor faults, respectively. Restarts of (parts of) the system are always classified as A faults, which should be solved immediately. Customers usually do not accept the system if any A faults occur during demonstration. C faults are typically aesthetic problems, the correction of which could eventually be postponed to the next release.

hours	week	A	B	C	total
161	1	1	2	0	3
229	2	5	2	3	10
176	3	5	3	3	11
304	4	6	3	3	12
225	5	3	1	2	6
585	6	7	13	6	26
432	7	2	6	9	17
2112	total	29	30	26	85

Table 8.5
Classification of software faults during ASV.

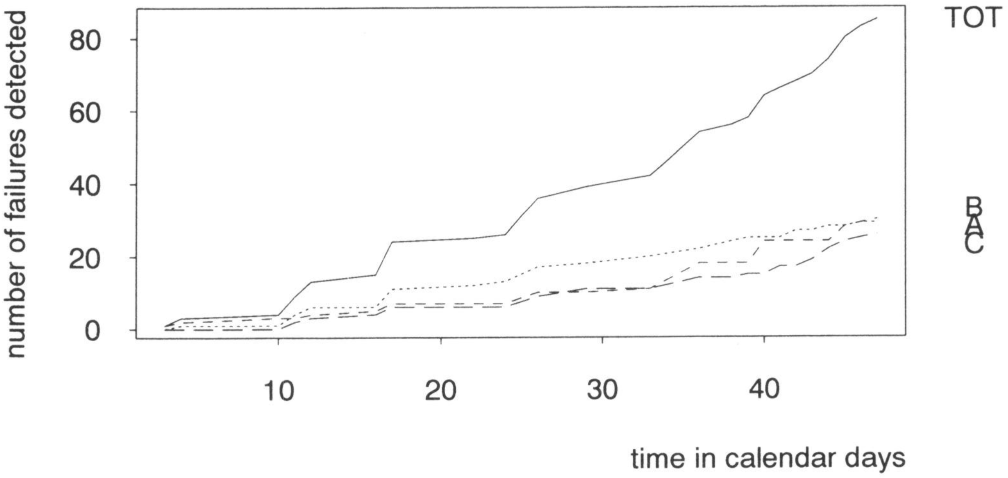


Figure 8.10
Number of faults detected against calendar time.

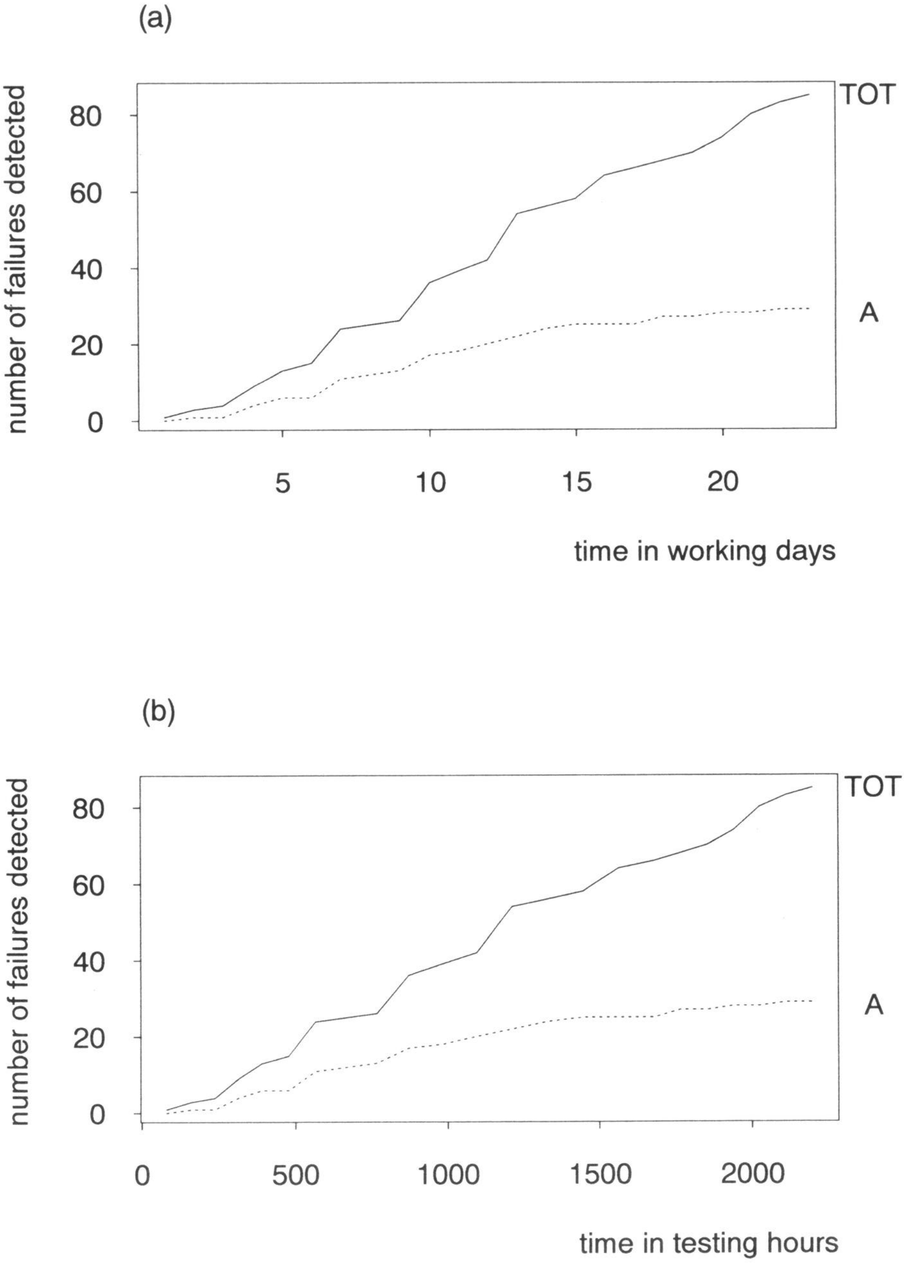


Figure 8.11
Number of faults detected against time in
(a) working days and (b) testing hours.

Cumulative numbers of A, B, C, and all faults detected against time in calendar days are plotted in Figure 8.10. We see that all curves seem to go up linearly in calendar time. In Figures 8.11a and 8.11b we plotted cumulative number of A and all faults against time in working days and time in testing hours, respectively. The two plots show no significant differences. The total number of faults still seems to grow linearly in time, while the curve for the A faults seems to grow negative exponentially. We might expect that software reliability theory could give reasonable insight in the behaviour of the software with respect to A faults. In the next section we will compare the Jelinski-Moranda (1972) and the Littlewood (1980) model and estimate their parameters. Concerning the linear growth of the total number of faults, one should have in mind, that as long as one is considering a piece of software, one will find ways to improve it a bit. In this respect it is not surprising that minor faults behave different from major ones.

8.4.3 Statistical inference

We will compute estimators for the model parameters using Maximum Likelihood Estimation. As in the PMS case-study we will not consider the data as grouped, but use the approach described in Section 8.2, estimating the test-intensity by the number of testing hours. Results are given in Table 8.6.

JM	total	A
$\hat{N}$	840±2885	46±19
$\hat{\phi}$	(0.47±1.70)e-04	(4.56±3.09)e-04
$\hat{\lambda}$	0.0357	0.0078
L	total	A
$\hat{N}$	230±163	34±6
$\hat{\alpha}$	(1.97±1.73)e-04	(7.86±3.29)e-04
$\hat{\varepsilon}$	0	0
$\hat{\lambda}$	0.0287	0.0039

Table 8.6
Parameter estimation for the models of
Jelinski-Moranda (JM) and Littlewood (L).

We see again -as was the case with the PMS data- that Littlewood's model suggests that the Jelinski-Moranda model is the best within the larger Littlewood class ($\hat{\varepsilon}$=0). Estimated curves for the two models are plotted in Figure 8.12. From this figure it becomes clear that both models give about an equal fit to the available data; prediction for future behaviour is, however, slightly different. In Figure 8.13 we compare the the estimated curves of the two models for A faults only and give 95% upper-bounds. We see that Littlewood's model not only gives slightly more optimistic predictions, but also has a smaller bandwidth.

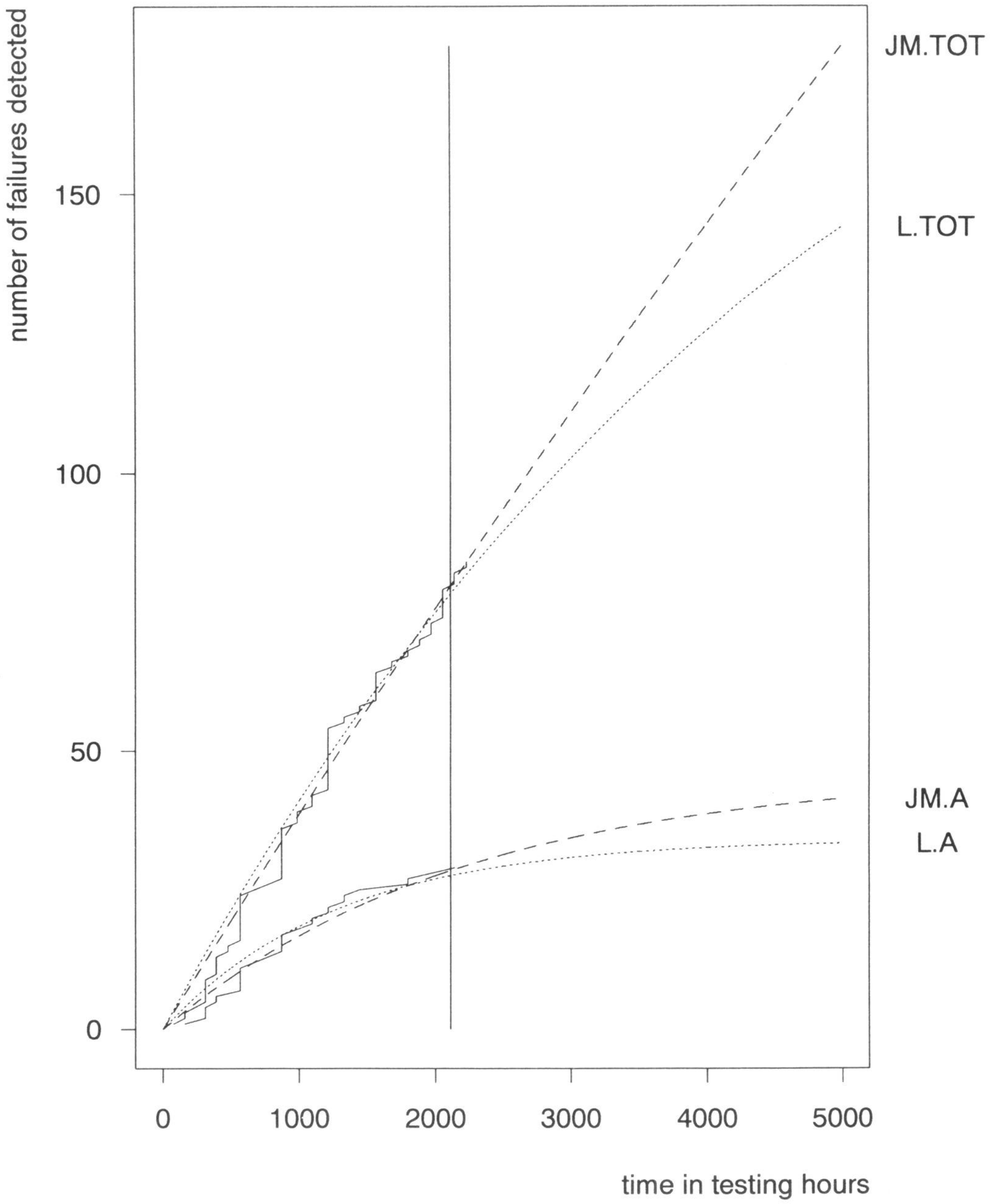


Figure 8.12
Estimates of mean value function for the models of
Jelinski-Moranda (JM) and Littlewood (L).

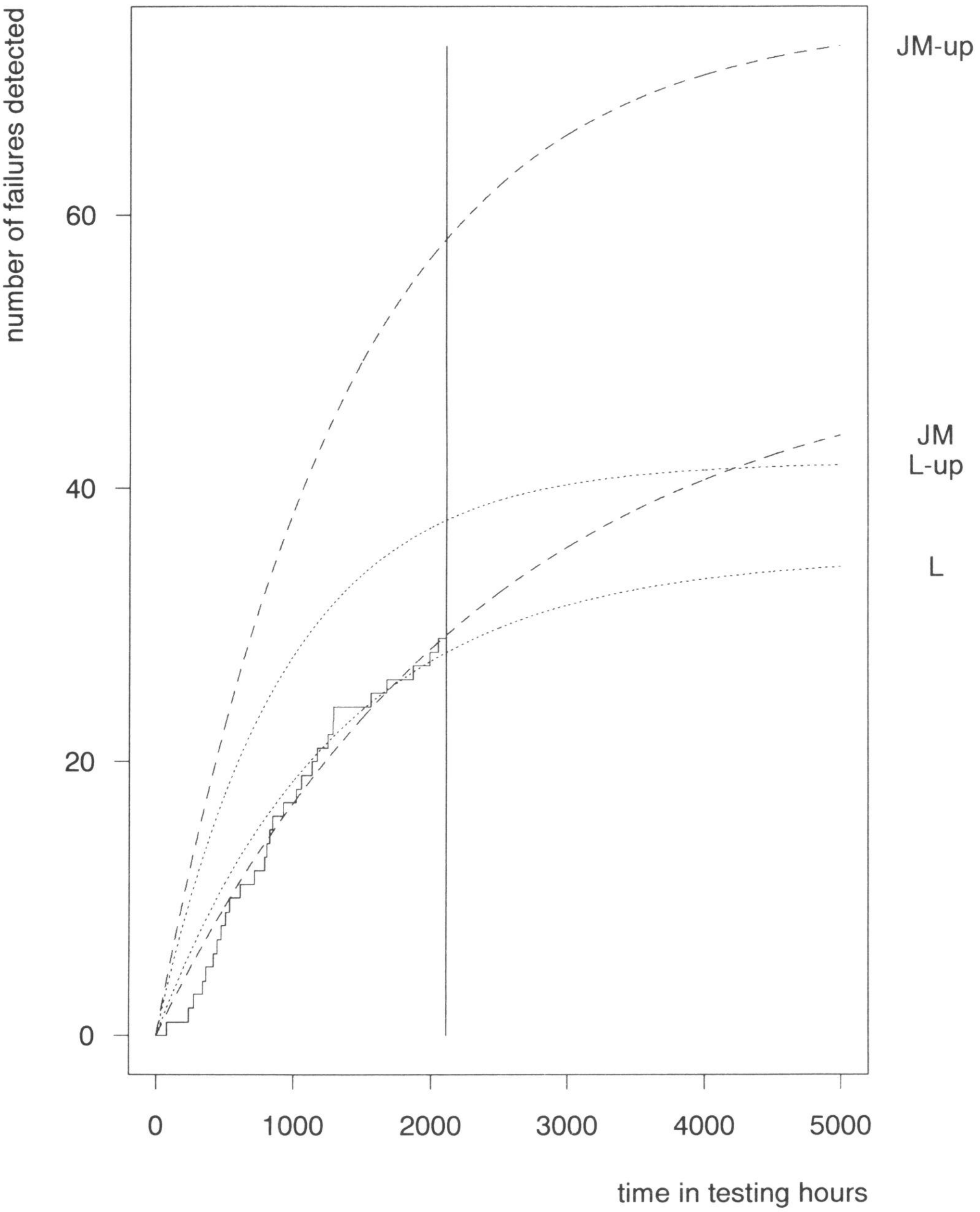


Figure 8.13
Estimates and 95% upper-bounds for the mean value function of priority A faults for the models of Jelinski-Moranda (JM) and Littlewood (L).

8.5 Problems, recommendations and concluding remarks

During the course of these case-studies we encountered several practical problems, related to collecting data in the real world of software development. These problems, which are likely to occur in practice with any software development group, in one way or the other obstructed a direct application of the developed software reliability theory. In this section we will mention the five most eye-catching ones, discuss the actions taken to reduce the negative effects they have on our software reliability results, and finally give recommendations how in our opinion the test process should be adapted in order that these problems will not appear at all in future.

(1) The test process is hardly or not automated. In software reliability theory we more or less take for granted that the test process is fully automated, which means that test inputs are generated, software bugs detected and all kinds of test data stored automatically. In practice, however, only a few software development companies have realised this kind of testing; more often conventional testing methods are used. In many software applications testing does not lend itself to automation. Based on what a tester perceives on a screen, he will push certain buttons and hence follow a certain test-path. Automatisation is here difficult to establish. There are many consequences of this. First of all, exact failure times of the error counting process will in general not be known but only the number of bugs detected per day (grouped data). This loss of information will lead to less accurate parameter estimations. Secondly, the stream of test data will be afflicted with larger inaccuracies than otherwise would be the case. Figures for test intensity or test effort per day will, if not registered in CPU time automatically, be liable to large subjectivities of individual testers. This will occur for the data concerning priority and subsegment of the faults too, not necessarily due to incapability or unwillingness of testers, but more often due to vagueness or ambiguities in definitions (Is a bug routine, urgent or very urgent, etcetera). The best way to overcome all this simply is to automate the whole test process. If this is not possible, one should pay a lot of attention to the data collection. Make clear agreements with the testers and avoid ambiguities in definitions. Finally, motivate the testing team to cooperate and make them aware of the fact that successful application of software reliability theory fully depends on the accuracy of the data.

(2) Testing is not random at all. Although this problem is related to the problem of automation (random testing cannot be performed without some automation of the test process), it is not a direct consequence of it and important enough to mention it here as a second problem. As described in Chapter 1 we assume in software reliability theory that in some sense software testing is a random process. It is sometimes wrongly understood, that for random testing all inputs should have equal probabilities of selection (uniform distribution). This is not true. We speak of random testing, if inputs are selected in a non-deterministic way with occurrence probabilities that coincide with the input distribution in the operational profile. Random testing takes place in practice only rarely. Besides an healthy aversion of management against any time-absorbing and inefficient looking methods, random testing can be difficult to put into practice, because the process cannot be automated (problem 1) or because no information about the input distribution in the operational profile is available. If one is testing by verifying the functionality of different subsegments one by one (which means that systematically from time to time new parts of the software are inspected) or if one is only using extraordinary inputs (with small occurrence probabilities), software reliability theory cannot be expected to give reasonable answers. During the alpha test at PMS this was the case. Therefore the alpha-test is not the most suitable test-period for applying software theory. More appropriate are the pi-test and

perhaps the beta-test. To make software reliability theory applicable we strongly recommend to spend at least a part of the available time and personnel to random testing. As input distribution one could ultimately choose the uniform distribution if all other information about the use in the field fails. If no proper random testing is possible but the number of tests is huge, one could decide to perform the specified tests in a random order (pseudo random testing). This will overcome a part of the problem. One should have in mind that if the tested inputs are not representative of the operational profile, translating the software reliability test results to practice can be very difficult.

(3) Hardware and software are closely related. In principle, we are investigating the behaviour of the software. Sometimes, however, only after closer examination of a system failure it is found out that it was not caused by a bug in the software but by one in the hardware. Hardware used during testing is often not exactly identical to the hardware used in the field. We do not want to take these hardware bugs into account. Another consequence of this strong relationship between software and hardware is that software test plans sometimes have to be changed because of hardware problems. This may lead to a decrease in test intensity (or better test efficiency), which is hard to measure precisely. It can also lead to a conflict with respect to the randomness of testing. Certain parts of the software are temporarily not inspected because of bugs in the hardware that interacts with this software. One should hence count only real software bugs and keep good track of when hardware is misbehaving. Try to quantify the test intensity (in testing hours per day) as accurate as possible.

(4) Minor faults show an occurrence behaviour which is completely different from that of major ones. The number of minor (aesthetical) faults detected tends to grow linearly in time, as occurrence of those kind of faults heavily depends on the available time to analyse and solve them. That is, the intensity of minor faults will be strongly influenced by the intensity of major faults, but not vice versa. Applying software reliability theory will therefore only make sense for major problems.

(5) Available time, computer facilities and personnel are limited. Finally, in practical applications everything turns on money: deadlines have to be kept, budgets have to be observed. This premise induces software managers to take actions and decisions which are in contradiction with the original software reliability theory assumptions. Such decisions are: (i) due to delays and deadlines, development and testing of parts of the software takes place simultaneously, (ii) alpha-test starts with too many bugs in the software, (iii) priorities of bugs are changed to speed up their solution. The second point is illustrated by the large number of bugs detected during alpha-test at PMS. This causes an enormous overhead (i.e. writing problem reports, updating database, weekly evaluation of new bugs, writing problem report answers, etcetera). The number of bugs found each day is therefore limited by the available manpower for testing and not by the quality of the software. Hence, in our opinion thorough testing takes place too late in the development phase to let the alpha-test be appropriate for application of the software reliability theory.

We have only mentioned five of the main problems to illustrate the complications arising when applying software reliability theory in practical situations. As stated earlier these problems are not typical for the software development at Philips or Ericsson, but are due to the complex process of software development itself, and to the many human factors involved.

Finally, we summarise our conclusions. The Software Reliability models and methods investigated so far can be applied to the data available at PMS and ETM, but the results are not very promising. The small number of field problems indicate that the released software is highly reliable. Because of the nature of the testing process and the quality of the test-data, it is not possible at this stage to use the results to provide the customer with strong reliability guarantees. The test process environment should be adapted (see the recommendations in the beginning of this section) in order to apply software reliability theory with more success. Of course there are also other (more simple, static) approaches to software reliability. We think of static models, Humphrey's regression method (Humphrey & Polson, 1992), etcetera. See also Section 1.4 of this thesis. Of course, also software reliability models and methods, as described in this thesis, should be further improved. We are, as Grady & Caswell (1987), strongly convinced that a thorough quantitative analysis of the software and of the software test process will lead to an increase of both efficiency and quality.

Appendix

A Software Reliability Tool in S-PLUS

The simulations and calculations presented in this thesis are performed in S-PLUS. S-PLUS is both a language and an interactive programming environment for data analysis and graphics. The S-PLUS language is a very high-level language for specifying computations. The language is part of an interactive environment: S-PLUS encourages you to compute, look at data, and program interactively, with quick feedback to enable you to learn and understand. For background information on S-PLUS we refer to Becker et al (1988).

During the course of this research project we have developed a package of S-PLUS functions which could be very useful when analysing the reliability of software. In this appendix we will demonstrate some of the features of this package in an informal way by playing a recorded S-PLUS session, analysing some simulated and real datasets. Hopefully, it will give the reader an idea of the strength and user-friendliness of S-PLUS as well. The ">" is the S-PLUS prompt character and the text following the "#" is an S-PLUS comment. All output of the S-PLUS programme is between quotes ("").

```
% Splus # From an Unix environment we call S-PLUS

"S-PLUS : Copyright (c) 1988, 1991 Statistical Sciences, Inc."
"S : Copyright AT&T."
"Version 3.0 Release 1 for Sun SPARC : 1991"
"Working data will be in /ufs/mark/.Data"

> cds("SWRtool") # Change directory to SWRtool
> ls() # List objects. Data-objects will have names starting with a "d", functions with an "f".

"d.real"
"f.bootstrap"
"f.confidence.intervals"
"f.create"
"f.goodness.of.fit"
"f.likelihood.contours"
"f.maximum.likelihood.estimation"
"f.plot"

> d.real # Show data-object d.real. This data-object is a list with two fields: [[1]] a character
# string with additional information and [[2]] a vector of failure times.

"[[1]]:"
"Real data set (Mellor & Littlewood)"

"[[2]]:"
" [1] 0.0020 0.0052 0.0066 0.0106 0.0146 0.0189 0.0231 0.0253 0.0286 0.0309
 [11] 0.0315 0.0323 0.0332 0.0337 0.0377 0.0446 0.0460 0.0533 0.0547 0.0578
 [21] 0.0628 0.0656 0.0719 0.0739 0.0740 0.0779 0.0792 0.0802 0.0810 0.0848
 [31] 0.0852 0.0863 0.0872 0.1021 0.1040 0.1043 0.1061 0.1087 0.1125 0.1178
 [41] 0.1180 0.1191 0.1192 0.1196 0.1212 0.1215 0.1220 0.1221 0.1260 0.1261
 [51] 0.1262 0.1291 0.1303 0.1331 0.1370 0.1375 0.1394 0.1406 0.1436 0.1437
 [61] 0.1452 0.1460 0.1464 0.1481 0.1482 0.1494 0.1496 0.1497 0.1504 0.1523
 [71] 0.1533 0.1534 0.1536 0.1538 0.1539 0.1552 0.1618 0.1713 0.1750 0.1767
 [81] 0.1773 0.1856 0.1871 0.1930 0.1963 0.1994 0.2015 0.2029 0.2040 0.2053
 [91] 0.2062 0.2141 0.2143 0.2152 0.2199 0.0225 0.2302 0.2304 0.2314 0.2323
[101] 0.2763 0.2864 0.2888 0.2959 0.3016 0.3017 0.3029 0.3048 0.3055 0.3057
[111] 0.3154 0.3156 0.3211 0.4036 0.4144 0.4188 0.4196 0.4207 0.4214 0.4296
[121] 0.4298 0.4300 0.4302 0.4379 0.4471 0.4569 0.4688 0.4821 0.4975 0.5013
[131] 0.5176 0.5295 0.5611 0.5687 0.5712 0.5744 0.5783 0.5834 0.6032 0.6323
[141] 0.6602 0.6761 0.6911 0.7036 0.7105 0.7461 0.7570 0.7596 0.7626 0.7655
[151] 0.8052 0.8133 0.8204 0.9009 0.9059 0.9220 0.9296 0.9605 0.9752 0.9928
[161] 1.0063 1.0139 1.0289 1.0375 1.0472"

> f.maximum.likelihood.estimation(d.real) # Compute MLE's for d.real.

"MLE's (and standard deviations) according to the JM model are:"
"[1] N.hat= 175 (4.9)"
"[2] phi.hat= 2.691 (0.307)"
"[3] log L (N.hat, phi.hat)= 1041.0"
"[4] lambdahat(tau)= 27.7"

"MLE's (and standard deviations) according to the LW model are:"
"[1] N.hat= 181 (3.978)"
"[2] alpha.hat= 2.970 (0.310)"
"[3] epsilon.hat= 0.666 (0.010)"
"[4] log L (N.hat, alpha.hat,epsilon.hat)= 709.8"
"[5] lambda.hat(tau)= 28.0"
```

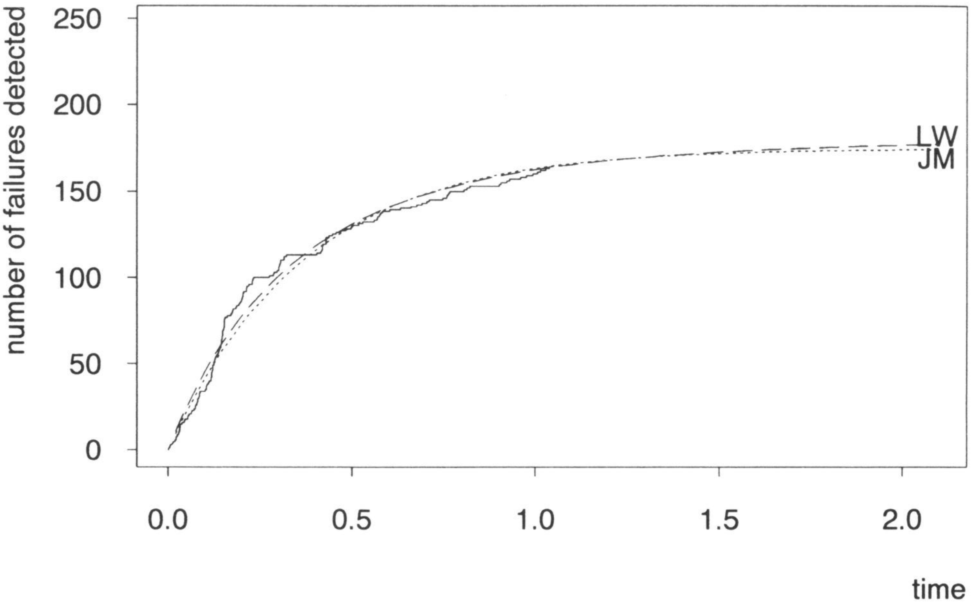


Figure A.1
Real data set (Mellor & Littlewood).

```
> d.jm<-f.create() # Create data-object d.jm.

"Simulated or real data?"
"1. Simulation of Jelinski-Moranda model"
"2. Simulation of Littlewood model"
"Your choice="1

"Initiate parameters of JM model:"
"N="100
"phi="1

"Choose stopping time:"
"tau="1

> d.jm  # Show data-object d.jm.

"[[1]]:"
"Simulation of Jelinski-Moranda model: N= 100, phi= 1, tau=1"

"[[2]]:"
" [1] 0.02862419 0.03525657 0.04673452 0.06128619 0.07283782 0.07303731
 [7] 0.08002712 0.08507235 0.10515043 0.11621881 0.11775297 0.15622693
[13] 0.15869000 0.17360860 0.18228630 0.18461936 0.19956877 0.20311595
[19] 0.20455902 0.20982772 0.21550167 0.25586825 0.26001808 0.27541685
```

```
[25] 0.27577482 0.27623442 0.30309317 0.31112298 0.31447566 0.32236892
[31] 0.32238472 0.32347081 0.32656447 0.33764846 0.35533056 0.37777466
[37] 0.42017139 0.44807096 0.46206174 0.46240867 0.47688924 0.49047940
[43] 0.50236441 0.50543355 0.51248065 0.51674496 0.52302515 0.52953694
[49] 0.54903581 0.54942346 0.55827129 0.57413293 0.60558100 0.60671885
[55] 0.65301759 0.69893211 0.69952784 0.70395771 0.71042998 0.71493852
[61] 0.74872407 0.74984765 0.77423497 0.78458072 0.80971732 0.95128294
[67] 0.99004575"
```

```
> f.maximum.likelihood.estimation(d.jm) # Compute MLE's for d.jm.

"MLE's (and standard deviations) according to the JM model are:"
"[1] N.hat= 95.6 (18.6)"
"[2] phi.hat= 1.22 (0.43)"
"[3] log L (N.hat, phi.hat)= 353.6"
"[4] lambda.hat(tau)= 34.0"

"MLE's (and standard deviations) according to the LW model are:"
"[1] N.hat= 85 (11.5)"
"[2] alpha.hat= 1.536 (0.449)"
"[3] epsilon.hat= 0.000"
"[4] log L (N.hat, alpha.hat,epsilon.hat)= 2159"
"[5] lambda.hat(tau)= 26.1"
```

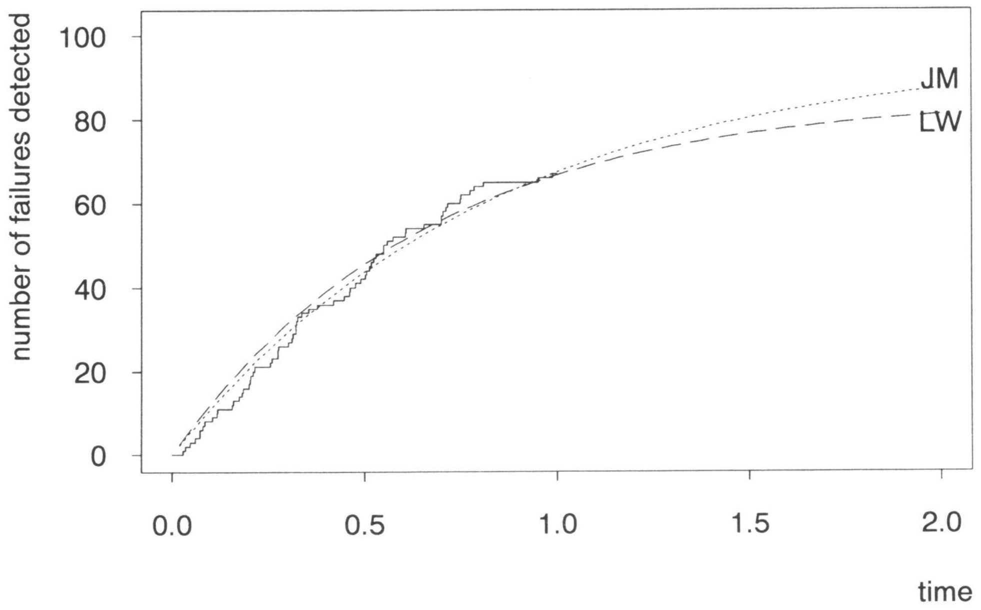


Figure A.2
Simulation JM model: $N=100, \phi=1$.

```
> f.confidence.intervals(d.jm) # Compute confidence intervals

"1. Jelinski-Moranda model"
"2. Littlewood model"
"Your choice="1

"Confidence intervals based on:"
"1. Approximate normal test statistic"
"2. Wilks likelihood ratio test statistic"
"3. Parametric bootstrap"
"Your choice="3

"1. One-sided  bootstrap"
"2. Two-sided  bootstrap"
"3. One-sided  studentized bootstrap"
"4. Two-sided  studentized bootstrap"
"Your choice="4

"Choose level of confidence:"
"p="0.90

"[1] 84 < N < 157"
"[2] 0.57 < phi < 1.97"

> f. likelihood.contours() # Inspect contour curves of likelihood function.
```

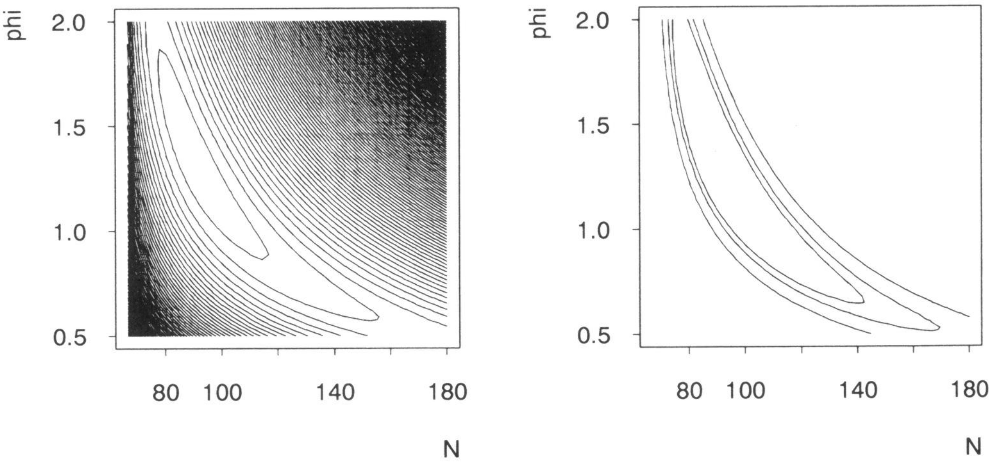


Figure A.3
Likelihood contour curves according to JM model.

```
> d.lw<-f.create() # Create data-object d.lw.

"Simulated or real data?"
"1. Simulation of Jelinski-Moranda model"
```

```
"2. Simulation of Littlewood model"
"Your choice="2

"Initiate parameters of LW model:"
"N="1000
"alpha="1
"epsilon="1

"Choose stopping time:"
"tau="4

> f.maximum.likelihood.estimation(d.lw) # Compute MLE's for d.lw.

"MLE's (and standard deviations) according to the JM model are:"
"[1] N.hat= 823 (6.6)"
"[2] phi.hat= 0.870 (0.040)"
"[3] log L (N.hat, phi.hat)= 5331.7"
"[4] lambda.hat(tau)= 22.2"

"MLE's (and standard deviations) according to the LW model are:"
"[1] N.hat= 1063 (15.2)"
"[2] alpha.hat= 1.014 (0.037)"
"[3] epsilon.hat= 1.365 (0.417)"
"[4] log L (N.hat, alpha.hat,epsilon.hat)= 3753.46"
"[5] lambda.hat(tau)= 41.7"

> f. likelihood.contours() # Inspect contour curves of likelihood function.
```

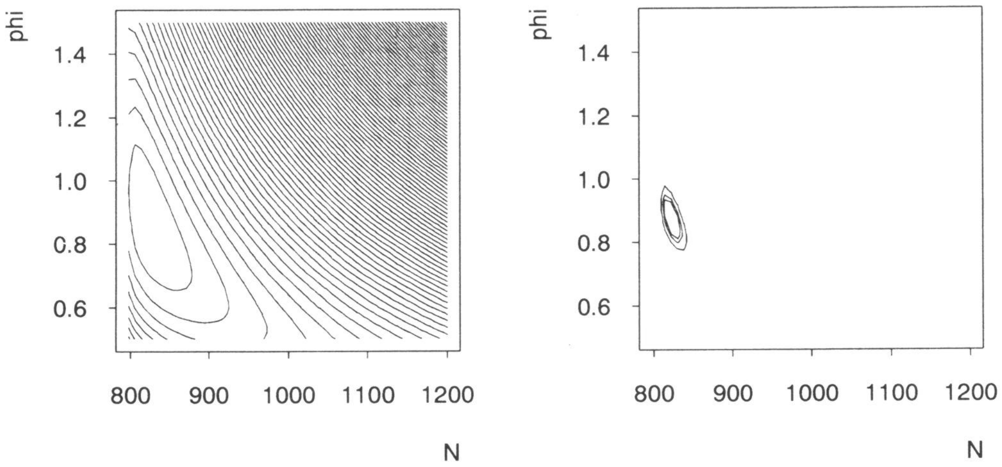


Figure A.4
Likelihood contour curves according to LW model.

```
> q() # End of S-PLUS session.
```

References

Aalen, O.O. (1978), Non-parametric inference for a family of counting processes, *Annals of Statistics* **6,** pp. 701-726.

Aalen, O.O. (1980), A model for non-parametric regression analysis of counting processes, *Lecture Notes in Statistics* **2,** pp. 1-25, Springer-Verlag, New-York.

Akiyama, F. (1971), An Example of Software System Debugging, *Information Processing 71,* North-Holland, New-York, pp. 353-359.

Allen, A.O. (1978), *Probability, Statistics and Queueing Theory,* Computer Science and Applied Mathematics, Academic Press, New York.

Andersen, P.K. & Borgan, Ø. (1985), Counting processes models for life history data: a review, *Scandinavian Journal of Statististics* **12,** pp. 97-158.

Andersen, P.K., Borgan, Ø., Gill, R.D. & Keiding, N. (1993), *Statistical Methods Based on Counting Processes,* Springer-Verlag, New York.

Anderson, T.W. (1955), The integral of a symmetric unimodal function over a symmetric convex set and some probability inequalities, *Proceedings of the American Mathematical Society* **6,** pp. 170-176.

Ascher, H.E. & Feingold, H. (1982), Repairable Systems Reliability: Future Research Topics, *Reliability in Electrical and Electronic Components and Systems,* North-Holland, Amsterdam, pp. 81-88.

Bahadur, R.R. (1967), Rates of convergence of estimates and some test statistics, *Annals of Mathematical Statistics* **38,** pp. 303-324.

Barendregt, L. & Van Pul, M.C. (1991), *An Analytical Solution to the Consistency Problem for the Littlewood Model in Software Reliability,* Report BS-R9129, Centre for Mathematics and Computer Science, Amsterdam.

Barendregt, L. & Van Pul, M.C. (1993), *Consistency of maximum likelihood and other estimators for the parameters of the Littlewood model in software reliability,* Unpublished paper (Submitted to Statistica Neerlandica).

Barlow, R.E. & Proschan, F. (1975), *Statistical Theory of Reliability and Life Testing Probability Models,* Holt, Rinehart and Winston, New York.

Basin, S.L. (1973), *Estimation of Software Error Rates via Capture-Recapture Sampling: A Critical Review,* Science Applications Report, Paolo Alto, CA.

Becker, R.A., Chambers, J.M. & Wilks, A.R. (1988), *The New S Language,* Wadsworth &Brooks/Cole Advanced Books & Software, Pacific Grove, CA.

Bickel, P.J. & Freedman D. (1981), Some asymptotic theory for the bootstrap, *Annals of Statistics* **9,** pp. 1196-1217.

Billingsley, P. (1961), *Statistical Inference for Markov Processes,* University of Chicago Press, Chicago.

Billingsley, P. (1968), *Convergence of Probability Measures,* Wiley, New York.

Borgan, Ø. (1984), Maximum likelihood estimation in parametric counting process models, with applications to censored failure time data, *Scandinavian Journal of Statistics* **1,** pp. 1-16. (corrigendum, *ibid.,* 275.)

Bremaud, P. (1977), Processes ponctuals et martingales: Résultats récents sur la modélisation et filtrage, *Advanced Applied Probability* **9,** pp. 362-416.

Carnahan, B. & Wilkes, J.O. (1973), *Digital Computing and Numerical Methods,* Wiley, New York.

Cramér, H. (1946), *Mathematical Methods of Statistics,* Princeton University Press, Princeton.

Dzhaparidze, K. & Valkeila, E. (1988), On the Hellinger type distances for filtered experiments, *Probability Theory and Related Fields* **85,** pp. 105-117.

Feuer, A.R. & Fowlkes, E.B. (1979), Some Results from an Empirical Study of Computer Software, *Proceedings Fourth International Conference on Software Engineering,* Munich, Germany, pp. 351-355.

Fletcher, R. & Reeves, C.M. (1964), Function Minimisation by Conjugate Gradients, *Computer Journal* **7,** pp. 149-154.

Gantmacher, F.R. (1954), *The Theory of Matrices,* Chelsea Publishing Company, New York.

Geurts, W.A.J., Hasselaar, M.M.A., & Verhagen, J.H. (1988), *Large Sample Theory for Statistical Inference in Several Software Reliability Models,* Report MS-R8807, Centre for Mathematics and Computer Science, Amsterdam.

Gilb, T. (1979), *Software Metrics,* Winthrop, Cambridge, p. 28.

Gill, R.D. (1980), *Censoring and Stochastic Integrals,* Mathematical Centre Tracts 124, Mathematisch Centrum, Amsterdam.

Gill, R.D. (1986), The total time on test plot and the cumulative total time on test statistic for a counting process, *Annals of Statistics* **14,** pp. 1234-1239.

Gill, R.D. (1989), *Non- and semi-parametric maximum likelihood estimators and the von Mises method (Part 1), Scandinavian Journal of Statistics* **16,** pp. 97-128.

Gill, R.D. & Johansen, S. (1990), A survey of product-integration with a view towards application in survival analysis, *Annals of Statistics* **18,** pp. 1501-1555.

Goel, A.L. and Okumoto, K. (1979), Time Dependent Error-Detection Rate Model for Software Reliability and other Performance Measures, *IEEE Transactions on Reliability* **28,** pp. 206-211.

Grady, R.B. & Caswell, D.L. (1987), *Software Metrics: Establishing a company-wide program,* Prentice-Hall, Englewood Cliffs, NJ.

Greenberg, M. (1966), *Lectures on Algebraic Topology,* W.A. Benjamin Inc., Menlo Park, California.

Hall, P. (1988), Theoretical comparison of bootstrap confidence intervals, *Annals of Statistics* **16,** pp. 927-953.

Halstead, M.H. (1977), *Elements of Software Science,* Elsevier, New York.

Helland, I.S. (1982), Central limit theorems for martingales with discrete or continuous time, *Scandinavian Journal of Statistics* **9,** pp. 79-94.

Helmers, R. (1991), On the edgeworth expansion and the bootstrap approximation for a studentised *U*-statistic, *Annals of Statistics* **19,** pp. 470-484.

Hjort, N.L. (1990), Goodness of Fit Test in Models for Life History Data Based on Cumulative Hazard Rates, *Annals of Statistics* **18,** pp. 1221-1258.

Humphrey, W. S. & Polson, N. (1992), *Projecting Software Defaults,* Unpublished paper.

Ibragimov, I.A. & Khas'minskii, R.Z. (1979), *Statistical Estimation - Asymptotic Theory,* Springer-Verlag, New York.

Jacod, J. (1975), Multivariate point processes: Predictable projection, Radon-Nikodym derivatives, representation of martingales, *Z. Wahrsch. verw. Geb.,* **31,** pp. 235-253.

Jacod, J. & Shiryaev, A.N. (1987), *Limit Theorems for Stochastic Processes,* Springer-Verlag, Berlin.

Jelinski, Z. & Moranda, P. (1972), Software Reliability Research, *Statistical Computer Performance Evaluation,* pp. 465-484.

Kardaun, O.J.W.F. (1986), *On Statistical Survival Analysis and its Application in Medicine,* PhD. Thesis, Groningen.

Kendall, M.G. & Stuart, A. (1967), *The Advanced Theory of Statistics,* Vol. II, 2nd Edition, Hafner Publishing Co., New York.

Khmaladze, E.V. (1981), Martingale approach in the theory of goodness of fit tests, *Theory of Probability and Applications* **16,** pp. 240-257.

Koch, G. & Sprey, P.J.C. (1983), A martingale approach to software reliability, *IEEE Transactions on Reliability* **32,** pp. 342-345.

Kulldorff, G. (1957), On the conditions for consistency and asymptotic efficiency of maximum likelihood estimates, *Skandinavisk Aktuarietidsskrift* **40,** pp. 129-144.

Kurtz, T.G. (1981), *Approximation of Population Processes,* Regional Conference Series in Applied Mathematics **36,** SIAM Philadelphia, Pennsylvania.

Kurtz, T.G. (1983), Gaussian approximations for Markov chains and counting processes, *Bulletin of the International Statistical Institute* **50,** pp. 361-375.

Lawless, J.F. (1982), *Statistical Models and Methods for Lifetime Data,* Wiley, New York.

Le Cam, L. (1990), Maximum likelihood: an introduction, *International Statistical Review,* **58,** pp. 153-171.

Le Cam, L. & Yang, G.L. (1990), *Asymptotics in Statistics,* Springer Verlag, New York.

Lenglart, E. (1977), Rélation de domination entre deux procèsses, *Annals Institute Henri Poincaré* **13,** pp. 171-179.

Littlewood, B. (1980), Theories of software reliability: How good are they and how can they be improved? *IEEE Transactions on Software Engineering* **6,** pp. 489-500.

Lloyd, D.K. & Lipow, M. (1977), *Reliability: Management, Methods, and Mathematics,* 2nd ed., published by the authors, Redondo Beach, CA.

Maglaperidze, N.O. & Tsigroshvili Z.P. (1993), *Goodness-of-Fit Tests for Testing Parametric Hypotheses about the Distribution of Point Processes,* Unpublished paper, Tbilisi.

Maglaperidze, N.O., Van Pul, M.C. & Tsigroshvili Z.P. (1993), *Goodness of Fit Tests for Statistical Models in Software Reliability,* Unpublished paper.

Mann, N.R., Schafer, R.E. & Singpurwallah, N.D. (1974), *Methods for Statistical Analysis of Reliability and Life Data,* Wiley, New York.

McCabe, T.J. (1976), A Complexity Measure, *IEEE Transactions on Software Engineering* **2,** pp. 308-320.

Mills, H.D. (1972), *On the Statistical Validation of Computer Programs,* IBM Federal Systems Division, Report FSC-72-6015, Gaithersburg, MD.

Moek, G. (1983), *Software Reliability Models on Trial: Selection, Improved Estimation, and Practical Results,* Report MP 83059 U, National Aerospace Laboratory NLR, Amsterdam.

Moek, G. (1984), Comparison of some software reliability models for simulated and real failure data, *International Journal of Modelling & Simulation* **4,** pp. 29-41.

Motley, R.W. & Brooks, W.D. (1977), *Statistical Prediction of Programming Errors,* Rome Air Development Center, Technical Report RADC-TR-77-175, Rome, NY.

Musa, J.D. (1975), A theory of software reliability and its application, *IEEE Transactions on Software Engineering* **3,** pp. 312-327.

Musa, J.D., Iannino, A. & Okumoto, K. (1987), *Software Reliability: Measurement, Prediction, Application,* McGraw-Hill Book Company, New York.

Nagel, P.M. & Skrivan, J.A. (1982), *Software Reliability: Repetitive Run Experimentation and Modelling,* Boeing Computer Service Company, BCS-40366, Seattle, WA.

Nagel, P.M., Scholz, F.W. & Skrivan, J.A. (1984), *Software Reliability: Additional Investigations into Modelling with Replicated Experiments,* Boeing Computer Service Company, BCS-40446, Seattle, WA.

Nelder, J.A. & Mead, R. (1965), A Simplex Method for Function Minimisation, *Computer Journal* **7,** pp. 308-313.

Pollard, D. (1989), *Convergence of stochastic Processes,* Springer Series in Statistics, New York.

Van Pul, M.C. (1990), *Asymptotic Properties of Statistical Models in Software Reliability,* Report BS-R9011, Centre for Mathematics and Computer Science, Amsterdam.

Van Pul, M.C. (1991a), *Software Reliability Simulations on the Musa Model,* Report BS-R9122, Centre for Mathematics and Computer Science, Amsterdam.

Van Pul, M.C. (1991b), *Modelling Imperfect Repair and Software Growth,* Proceedings, Interface '91, Seattle.

Van Pul, M.C. (1991c), *Software Reliability and the Bootstrap,* Report BS-R9123, Centre for Mathematics and Computer Science, Amsterdam.

Van Pul, M.C. (1992a), Asymptotic Properties of Statistical Models in Software Reliability, *Scandinavian Journal of Statistics* **19,** pp. 235-253.

Van Pul, M.C. (1992b), Simulations on the Jelinski-Moranda Model of software reliability; application of some parametric bootstrap methods, *Statistics and Computing* **2,** pp. 121-136.

Rao, C.R. (1973), *Linear Statistical Interference and its Applications,* Wiley, New York.

Rebolledo, R. (1980), Central limit theorems for local martingales, *Z. Wahrsch. Verw. G.* **51,** pp. 269-286.

Rook, P. (1990), *Software Reliability Handbook,* Elsevier Applied Science, London.

Rudner, B. (1977), *Seeding/Tagging Estimation of Software Errors: Models and Estimates,* Rome Air Development Center, Technical Report RADC-TR-77-15, Rome, NY.

Shooman, M.L. (1968), *Probabilistic Reliability: An Engineering Approach,* McGraw-Hill, New York.

Shorack, G.R. & Wellner J.A. (1986), *Empirical Processes with Applications to Statistics,* Wiley, New York.

Singh, K. (1981), On asymptotic accuracy of Efron's bootstrap, *Annals of Statistics* **9,** pp. 1187-1195.

Sunohara, T., Takano, A., Uehara, K. & Ohkawa, T. (1981), Program Complexity Measure for Software Development Management, *Proceedings Fifth International Conference on Software Engineering,* San Diego, pp. 100-106.

Thayer, T.A. (1976), *Software Reliability Study,* Rome Air Development Center, Technical Report RADC-TR-76-238, Rome, NY. Seattle, WA, pp. 106-109.

Van der Vaart, A.W. (1987), *Statistical Estimation in Large Parameter Spaces,* Mathematical Centre Tracts 44, Mathematisch Centrum, Amsterdam.

Vetterling, W.T., Teukolsky, S.A., Press, W.H. & Flannery, B.D. (1985), *Numerical Recipes: The art of Scientific Computing,* Cambridge University Press, Cambridge.

Wald, A. (1949), Note on the consistency of the maximum likelihood estimate, *Annals of Mathematical Statistics* **20,** pp. 595-601.

Samenvatting

Computersystemen zijn onmisbaar geworden in onze huidige samenleving. Soms is de betrouwbaarheid van hard- en software van vitaal belang voor de gebruikers ervan. Denk aan toepassingsgebieden als lucht- en ruimtevaart, defensie of gezondheidszorg. Het moge evenwel duidelijk zijn, dat ook bij de ontwikkeling van minder delicate computersoftware een hoge mate van *betrouwbaarheid* gewaarborgd dient te worden aan de klant. Uiteraard beweert elke software-firma dat bij het ontwikkelen zo gestructureerd te werk gegaan is, dat de software correct is. Aangezien de logische complexiteit van software in het algemeen veel groter is dan die van bijvoorbeeld hardware, is het *correct bewijzen* van software meestal een onmogelijke opgave. In de praktijk blijkt ook, dat software -na te zijn "voltooid"- nog vol zit met fouten. Software wordt daarom geruime tijd en intensief getest, alvorens vrijgegeven te worden. Hier ontstaat een moeilijke afweging van kosten en levertijd aan de ene kant en kwaliteit aan de andere. Immers, niet alleen kost het testen van software handenvol geld, bovendien bestaat het risico dat de concurrent hetzelfde product iets eerder op de markt brengt. Aan de andere kant zullen de verkoopcijfers van een onbetrouwbaar product tegenvallen en kan zelfs het imago van de firma aangetast worden. Het is dus uitermate zinvol na te gaan hoe de betrouwbaarheid van computersoftware tijdens de test- en ontwikkelfase verandert, of aan bepaalde betrouwbaarheidscriteria reeds voldaan is, of hoe lang nog verder getest dient te worden. Binnen de wiskunde en computerwetenschappen bestaat een jonge discipline, die zich met dit soort vragen bezighoudt en er zowel kwalitatieve als kwantitatieve antwoorden op tracht te geven: de *Software Reliability.*

Bij de modellering van software reliability kunnnen verschillende paden bewandeld worden. Veel onderzoekers houden zich bezig met de zogenaamde statische complexiteits analyse. Dit houdt in dat het aantal fouten in een stuk software gemodelleerd wordt in termen van bepaalde softwarekarakteristieken, zoals de lengte van het programma, de taal waarin het geschreven is, het aantal functie aanroepen, etcetera. In dit proefschrift behandelen wij een andere, meer dynamische aanpak, namelijk die van de *fout-tellende modellen* (error-counting models). Dit zijn modellen, die uitgaan van het volgende test-experiment. Een computerprogramma wordt gedurende een vooraf gespecificeerde tijd τ geëxecuteerd. Inputs worden aselect gekozen uit de inputspace die representatief geacht wordt voor het gebruik in de praktijk. Vervolgens produceert het programma hetzij de correcte output, of een software failure treedt op. Dat wil zeggen, de software produceert een verkeerd antwoord of helemaal geen output. Na de ontdekking van een failure, wordt de CPU-clock stopgezet en het programma naar een team van debuggers gestuurd. Als dit team, de software fault opgespoord en verwijderd heeft, worden gegevens betreffende de failure en de fault (voor zover beschikbaar) opgeslagen in een databestand. Hierna wordt de CPU-clock weer gestart en de testprocedure wordt hervat met een nieuwe input, totdat de executieperiode τ verstreken is. Deze error-counting models zijn dynamische modellen, die zich alleen baseren op bepaalde testdata (zoals de tijdstippen waarop fouten optreden). De modellen hangen dus niet expliciet af van factoren als de lengte en structuur van het programma, de taal waarin het geschreven is, de vaardigheid van de tester, enz. Door de informatie te gebruiken, die we verkrijgen uit het experiment, kunnen we de parameters van het onderliggende model schatten; in het bijzonder het totaal aantal fouten initieel aanwezig in de software. Bepaalde functies van de modelparameters zullen ons schattingen voor interessante grootheden (zoals de failure intensity, de reliability, de mean time between failures en de release time) opleveren. Sinds begin jaren zeventig zijn er tientallen error-counting models gelanceerd, elk gebaseerd op iets andere veronderstellingen over het testproces. Soms nemen we aan, dat fouten in de software onafhankelijk van elkaar zullen optreden en dat op het moment dat een fout ontdekt wordt, deze onmiddellijk wordt verbeterd zonder dat hierbij nieuwe fouten worden geïntroduceerd (perfect repair). Dit is het geval bij drie zeer bekende EC&D modellen: het *Jelinski-Moranda (JM) model,* het *Goel-Okumoto (GO) model,* en het *Littlewood (L) model.*

Niet alleen de keuze van het model stelt ons voor verschillende afwegingen. Ook de schatting van de parameters in het gekozen model is een probleem op zich. Als schattingsmethode zullen we meestal de *maximum likelihood estimation procedure* gebruiken. We leiden een expressie af voor de kans af op de geobserveerde data als functie van de parameters. Deze functie, die de *likelihood* genoemd wordt, willen we maximaliseren. Afleiding van de likelihood-functie is niet altijd analytisch mogelijk en de numerieke bepaling van zijn maxima kan instabiel zijn. Onder redelijke voorwaarden kan echter aangetoond worden, dat de maximum likelihood estimation procedure bruikbare schatters oplevert met aantrekkelijke (asymptotische) eigenschappen.

Dit proefschrift bestaat uit drie delen. In het eerste inleidende deel wordt een algemene introductie tot de software reliability theorie gegeven en behandelen we de belangrijkste concepten uit de telproces en martingaal theorie, die we in het tweede deel nodig zullen hebben. Het tweede deel bevat de theoretische resultaten van dit proefschrift en bestaat uit vijf afzonderlijke publicaties. In hoofdstuk 3 bestuderen we de asymptotische eigenschappen van de maximum likelihood estimator (MLE) voor een ruime klasse van modellen. In hoofdstuk 4 bewijzen we de validiteit van de parametrische bootstrap voor deze zelfde klasse. In hoofdstuk 5, worden voor het Littlewood model verscheidene oplossingen gepresenteerd en vergeleken voor het klassieke probleem van meerdere wortels van het stelsel likelihood equations. In hoofdstuk 6 onderzoeken we de limietverdeling van een generalisatie van de Kolmogorov-Smirnov goodness of fit test statistic in onze situatie, terwijl we in hoofdstuk 7 een nieuw (imperfect repair) model introduceren en zijn statistische eigenschappen analyseren. Als toepassing van de software reliability theorie worden in het derde deel case-studies bij Philips Medical Systems en Ericsson Telecommunications besproken. We besluiten dit proefschrift met een appendix, waarin we een informele beschrijving geven van een software reliability tool geïmplementeerd in de statistische programmeeromgeving S-PLUS.

Curriculum Vitae

Mark van Pul werd op 30 april 1966 geboren te Oss. In 1984 behaalde hij het diploma Gymnasium β met Duits als extra vak aan het Titus Brandsma Lyceum te Oss, waarna hij zijn studie wiskunde aan de Rijksuniversiteit Utrecht begon. Zijn belangstelling voor toepassingen van de wiskunde bracht hem er toe ook de propedeutische examens in de informatica en in de natuurkunde af te leggen. Tijdens zijn doctoraalstudie koos hij de variant Toegepaste Research met een groot bijvak informatica. Na een afstudeerfase in de wachtrijtheorie bij Prof. Dr. Ir. J.W. Cohen behaalde hij op 23 januari 1989 zijn bul. Vanaf maart 1989 was hij gedetacheerd op het Centrum voor Wiskunde en Informatica te Amsterdam en als onderzoeker in opleiding verbonden aan het STW-project "Statistical Analysis of Error-Counting and Debugging Models in Software Reliability" onder supervisie van Prof. Dr. R.D. Gill. Zoals bij STW-projecten gebruikelijk is ook de toepasbaarheid van de resultaten van het onderzoek getoetst. Dit gebeurde middels case-studies bij Philips Medical Systems (Best) en Ericsson Telecommunicatie Maatschappij (Rijen).

Stellingen

behorende bij het proefschrift van

Mark C.J. van Pul

Statistical Analysis of Software Reliability Models

I

Software-ontwikkelaars staan over het algemeen sceptisch tegenover software reliability onderzoek. Alleen een software equivalent van de Chernobyl-, Bhopal- of Bijlmer-ramp lijkt hierin verandering te kunnen brengen.

II

De condities voor Stelling 3.1 uit dit proefschrift kunnen worden afgezwakt: differentieerbaarheid tot de derde orde (van β naar θ) is een klassieke, eenvoudig te verifiëren, doch onnodig strenge eis. Toegepast wiskundigen dienen evenwel bij het afzwakken van condities de verifieerbaarheid ervan niet uit het oog te verliezen.

III

Lemma 7.1 uit dit proefschrift laat zich — in plaats van inductief analytisch — ook direct met probabilistische argumenten bewijzen.

IV

De door de theorie aanbevolen één-staps Newton-Raphson methode werkt in de praktijk vaak slecht.

V

Mensen die — net als de paus — menen dat sexualiteit alleen bedoeld is voor de voortplanting, kunnen beter — net als de paus — kinderloos blijven.

VI

Werkelijk liefhebben is het vinden van het juiste evenwicht tussen je betrokken voelen met- en ruimte geven aan de ander. Bij moeder-kind relaties is het zoeken naar dit evenwicht vaak een moeizaam proces.

VII

Eenheid onder mensen kan alleen ontstaan, wanneer de verschillen die er zijn, geaccepteerd worden.

VIII

De meeste files op de A2 zijn als *langzaamrijdend* te classificeren. Files op de A1 komen vaker in aanmerking voor het predicaat *stilstaand,* vooral als de brug bij Muiden openstaat.

IX

Gezien de op handen zijnde afschaffing van de militaire dienstplicht is het de vraag of de oude wapenspreuk *"Je maintiendrai"* gehandhaafd kan blijven.

X

Bij de geboorteaangifte heeft de ambtenaar van de burgerlijke stand bij het inschrijven van mijn tweede voornaam in het geboorteregister — voor wat de spelling betreft — wellicht getwijfeld tussen *Cornelis* en *Cornelius.* Wetende dat volgens het Burgerlijk Wetboek spellingvarianten bij voornamen zijn toegestaan, maakte hij er *Cornelus* van. Een goede ambtenaar weet keuze's altijd te omzeilen.