Performance and Dependability Modelling with Second Order Fluid Stochastic Petri Nets

Katinka Wolter

Performance and Dependability Modelling with Second Order Fluid Stochastic Petri Nets

vorgelegt von
Diplom-Informatikerin
Katharina Wolter

Vom Fachbereich 13 - Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktorin der Naturwissenschaften
- Dr.rer.nat. -

genehmigte Dissertation

Promotionsausschuß:

Vorsitzender: Prof. Dr.-Ing. Stefan Jähnichen
Berichter: Prof. Dr.-Ing. Günter Hommel
Berichter: Prof. Dr. Kishor S. Trivedi
Berichter: Prof. Dr.-Ing. Adam Wolisz

Tag der wissenschaftlichen Aussprache: 27. August 1999

Berlin 1999

D 83

Abstract

Second order fluid stochastic Petri nets are a formalism to be used for performance and dependability modelling of technical systems. Second order FSPNs have a discrete and a continuous model part, both having a stochastic nature. Therefore, the formalism is especially suitable for the probabilistic modelling of hybrid systems. Other application fields are the modelling of reward measures that evolve continuously over time or the approximation of large streams, that are processed following a non-exponential distribution.

The discrete part of a second order FSPN is equivalent to a generalized stochastic Petri net (GSPN), whereas the continuous part consists of fluid places that are filled with a continuous amount of fluid, which flows along double-lined fluid arcs at normally distributed rate. Furthermore, a certain amount of fluid may be removed or deposited at once. This only happens along thin, single-lined, arcs. With the instantaneous removal or depositing of fluid the fluid level undergoes an instantaneous jump up or down. The incorporation of jumps in the formalism as well as regarding second order fluid models as opposed to first order models are the main contribution of this thesis.

Second order FSPNs extend known FSPNs in so far as they allow for a variability in the fluid flow. In this thesis, second order FSPNs with jumps have been formally defined and their semantics has been specified. The stochastic process that underlies a second order FSPN is a Markov process in continuous time with mixed continuous and discrete state space. If the model includes jumps, the equations are similar to the integro-differential equation of Takács. Otherwise they are second order partial differential equations.

A numerical method has been developed for the stationary and the transient solution of second order FSPNs with one and two fluid places. The transient analysis has been implemented and applied to examples. If possible the fluid models have been compared with discrete ones, in order to verify the formalism as well as the solution components.

Zusammenfassung

Fluide stochastische Petri-Netze zweiter Ordnung dienen der Leistungs- und Zuverlässigkeitsanalyse technischer Systeme. FSPNs zweiter Ordnung haben einen diskreten und einen kontinuierlichen Modellteil, die beide stochastisches Verhalten aufweisen. Daher eignet sich der Formalismus insbesondere auch zum probabilistischen Modellieren hybrider Systeme. Weitere Anwendungsgebiete sind die direkte Modellierung von Leistungsmaßen, die sich kontinuierlich über der Zeit entwickeln und die Approximation großer Ströme, die gemäß einer nicht-exponentiellen Verteilung verarbeitet werden.

FSPNs zweiter Ordnung erweitern die bekannten FSPNs insofern, als sie einen stochastisch variierenden fluiden Fluß ermöglichen. In dieser Dissertation werden fluide stochastische Petri-Netze zweiter Ordnung, die auch Sprünge enthalten können formal definiert und ihre Semantik wird festgelegt. Der stochastische Prozeß, der einem FSPN zweiter Ordnung zugrunde liegt, ist ein Markow-Prozeß in kontinuierlicher Zeit mit gemischt kontinuierlichem und diskretem Zustandsraum. Falls das Modell Sprünge im kontinuierlichen Modellteil enthält sind die zugrunde liegenden Gleichungen der Integro-Differentialgleichung von Takács ähnlich, sonst handelt es sich um partielle Differentialgleichungen zweiter Ordnung.

Zur Lösung partieller Differentialgleichungen zweiter Ordnung gibt es keine Standardverfahren. Ein numerisches Lösungsverfahren für die transiente und die stationäre Analyse von Modellen mit bis zu zwei fluiden Stellen wurde entwickelt.

Das Verfahren für die transiente Analyse ist implementiert und wird für die Untersuchung von beispielhaften Anwendungen verwendet. Wo möglich werden diskrete Modelle zum Vergleich herangezogen. Der Vergleich dient sowohl der Verifikation des Formalismus als auch der Lösungskomponenten.

Acknowledgments

Many people have entered my life while I was working on this thesis. At this place I would like to mention some of them in particular.

First I want to express my thanks to Günter Hommel for his support during the last years and for tolerating my rather unusual office hours over the last few months.

The German Research Council (DFG) has supported my work financially, first with a grant in the graduate course on Communication-Based Systems and later with a position in a research project. The biannual meetings of the graduate course were an opportunity to learn about other views on communication based systems. Moreover have they been occasions for wonderful questions and remarks by Adam Wolisz that suddenly made queueing theory become a lively affair.

I would like to thank Kishor Trivedi for his invitation to Duke University. His positive attitude towards problems that seem unsolvable, as there are many in the field of FSPNs, has been highly motivating.

Thanks to Reinhard German and Armin Zimmermann I very quickly felt at home in our group. Reinhard initiated this work and answered a large number of questions over the first months. He in some sense 'put me on the tracks'. Armin often found the time for a chat and hence participated in early design decisions.

The numerical solution component for the transient analysis is available thanks to the efforts of Andrea Zisowsky, who never complains when I still bug her to help me add extensions to the program. Olga Guimpeliovskaia is currently including the numerical algorithms in TimeNET. Eveline Homberg and Tobias Zepter made some of the drawings. Gudrun Pourshirazi has always been most welcoming to Louise, so I could make appointments at any time in the day. That, indeed, was a great help.

Meeting my room mate Armin Heindl every day makes work much more enjoyable. I would not want to miss neither our many technical discussions nor the conversations on every day's matters. So, even his discussions with Igor Kovalenko had an influence on my work.

Aad van Moorsel has been a human encyclopedia of performability modelling with extremely short response time. Even more, he has been a very good friend.

Finally, I must mention Louise, who has always been the perfect daughter of a working Mum and Tobias Zepter, who spent much time with her over the last weeks. Thanks for coming home every afternoon, although you sometimes were a bit late. And thanks for cooking every evening, Louise and I appreciated it very much.

The last sentence is for my grand-parents.

Contents

Chapter 1

Introduction

When planning or constructing a technical system, its properties and behaviour are of interest even before the system can actually be observed. In other cases the output of e.g. a manufacturing system is not as desired, but nobody knows why. These are typical situations when abstract, formal or mathematical modelling comes into play.

With the development of more and more complex technical systems performance and reliability evaluation becomes increasingly important. Neither can in safety critical systems, like avionic or nuclear systems, the risk of trial and error be taken, nor in large expensive manufacturing or telecommunication systems the cost of just trying to build an economic and effective system. Typically this involves constructing and solving some kind of a model. Hence, modelling formalisms are needed that are able to capture increasingly large and complex systems.

Petri nets are a very popular and powerful formalism for the modelling of complex technical systems. This thesis deals with fluid stochastic Petri nets, extensions to which will be made. Therefore here, we will restrict ourselves to giving an introduction to Petri nets and a brief survey of their history, after having explained in the first section their intended use.

Fluid models can either be used for the modelling of continuous system components, or a fluid stream is used for the approximation of items that are processed in a non-exponentially distributed activity. In a later section the contribution, that shall be made with this thesis in both fields shall be discussed. The last section of this chapter will give an outline of the remainder of this thesis.

Stochastic Petri nets can be evaluated in different ways. They can be solved analytically or simulated. Simulation leads to estimates for performance and reliability measures and their confidence intervals. Since simulation will not be used for the fluid models, it will neither be regarded in general. Instead, the analytical formulation of the dynamics of a second order FSPN will be derived and the numerical solution of the obtained differential equations will be studied in detail.

1.1 Performance and Reliability Modelling

There are many ways to regard and evaluate technical systems, depending on the point of view that is taken. We will mention only two possible points of view, that is the user's perception and the system view. The user will be interested foremost in the product or service he/she receives and more precisely in its amount, the performance of the system. Whereas when regarding the system itself, it is of main interest, whether its modules or components are working well or not, that is the reliability of the system and its modules.

Since neither the user demand, not the status of the system components is static or predictable, stochastic models are needed for both performance and reliability evaluation. The reliability evaluation, that is the probability of a service being delivered, or the probability of a system being operational, is called dependability.

The probabilistic model for performance evaluation represents variation in the internal state, like the number of jobs being served or waiting at each module, and variation in the external state, like the number of jobs arriving to the system. In contrast, the probabilistic model for reliability evaluation represents variation in the structure of the system (for each type of resource the number of working units) while ignoring internal and external state.

When regarding fault-tolerant systems, which consist of several redundant modules, dependability studies can become very complex. In simple words, modules can fail and be repaired. But both failure and repair should be regarded more in detail. Faults can be permanent, so they need to be detected and then removed. The coverage denotes the probability of being able to recover from a fault that has occurred. Faults can also be transient, which means they occur, last for a certain time and then the system is operating correctly again. The system consists of working and spare modules, both of which can fail with different consequences and chances of being detected. For a detailed discussion of failure and repair types, their detection and removal see [Lap92], reliability models for fault-tolerant systems have been studied in [NA80].

If performance is not degradable, that means if the system is either operating or not it is reasonable to regard performance and dependability in separate models. If on the contrary, performance is degradable, the system can operate on different levels of performance with a different number of modules that are working correctly. In this case, the total performance of the system can only be obtained by solving as many models as there are performance levels. Then it is much more efficient to combine the two to a *performability* model [Mey80, MFW80].

For the dependability model a finite state stochastic process is used, which very often is a Markov process, where each state represents a combination of working and failed resources, whereas performance as such is just a random variable with a probability distribution, that depends on the system configuration. When combining them, a performability model is obtained, that consists of the dependability model and performance measures are incorporated by associating a reward rate with every state of the stochastic process

[Mey82, STR88]. Hence, the total performance is computed as the sum of the state probabilities multiplied with the reward rate that is connected to that state. Apparently the challenge in solving such a model is solving the stochastic process.

Performance is not necessarily measured in the time the system has spent in one state of production or service, it can also mean that certain actions in the system or transitions from one state to another have taken place. This leads to the definition of *impulse rewards* as opposed to *rate rewards* [SM91].

Performance and reliability modelling is not only applied to fault-tolerant computer and manufacturing systems but also to telecommunication systems. While the interest in the former lies in security, safety and availability, in the latter the central issue is Quality of Service (QoS), which comprises waiting time, throughput and loss rate, all of which are sub aspects of performance and dependability. For an elaborate discussion and many references see [vM94].

Due to many application areas of performance and reliability it is desirable to make the methods developed available to a broad public. Performance and reliability modelling consists in formulating a stochastic process and reward measures. The stochastic process has to be solved and the reward measures must be evaluated [QSvMG96, dSG92]. The construction of a stochastic process is not always an easy and quickly to learn task. Therefore many abstract and graphical formalisms have been developed and implemented in software tools, that are meant to make the construction of the stochastic process easier and more intuitive. Most of them are extensions of Petri nets. But also stochastic process algebras have been used [DRH95]. Both are directly connected to a stochastic process, which is hidden from the user, as are the solution algorithms. Some formalisms have incorporated constructs, that allow a convenient definition of sophisticated reward measures, like UltraSAN [SDQW95, CFJ+91], some focus on powerful solution algorithms for complex stochastic processes [GKZH95, Hav95], and some have different modelling formalisms incorporated besides stochastic Petri nets, such as queueing systems, fault-trees, like SHARPE [STP95]. The first tool for GSPNs was GreatSPN [Chi92]. Parts of it have been used in early stages by many of the others.

Most of nowadays' research in performance and dependability analysis is focussed on the development of solution algorithms for complex stochastic processes. The second very important field is the definition and computation of reward measures, such as complete distributions as opposed to only their moments.

1.2 A Brief History of Petri Nets

This section will give a concise survey of the history of Petri nets, emphasizing the branches that influenced the invention of fluid stochastic Petri nets, the class of Petri nets (PN) that is central in this thesis. The connections are sketched in figure 1.1. The solid arrow connect model classes, that directly followed from one another, whereas the dotted arrows denote an influence and similarity, but not a subclass relation.

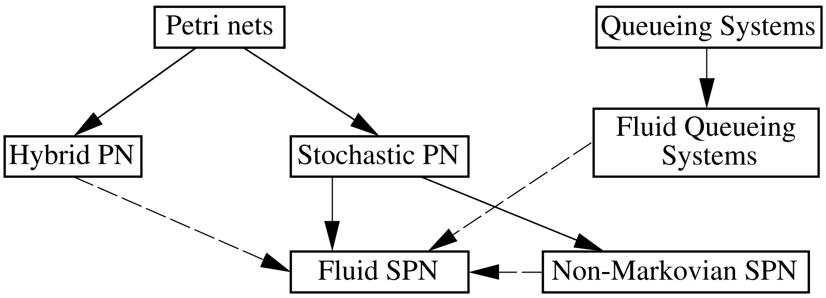

Figure 1.1: Structure of Petri net classes

All the concepts from figure 1.1 will be discussed in this thesis, the queueing systems in chapter 3 and the Petri net classes in this section.

Petri nets have been invented by Adam Petri in 1962 [Pet62]. They were meant to be used for the formal and graphical representation of concurrent systems. They consist of places, transitions, arcs connecting both and tokens. Places are drawn as circles and represent the passive parts of a system, like buffers, pipes, or queues, whereas transitions are drawn as rectangles and represent the activities in the system such as failure, repair and the processing of items. The items are represented by the tokens, drawn as small black dots, that are moved from one place to another by the firing of the transitions, along the arcs connecting places and transitions. Petri nets are well suited for modelling synchronisation and concurrency in distributed systems.

Regarding a systems properties we distinguish qualitative and quantitative properties. The qualitative ones are determinism (does a system for a given input always give the same output), correctness (does a system give the wanted, or specified output for a given input), deadlock-freeness (is it impossible, that situations arise, where all system components are waiting for input from another system component in order to be able to continue their service) and others more. These questions can be answered with models in many different formalisms (process algebras, Petri nets, etc.). Quantitative properties are performance (how much of a certain service or product will a system provide in a certain time period) and reliability (how likely is the system to provide its service and not to break down, either partially or completely), as discussed in more detail in the previous section. Petri nets are used in both fields. But different types of Petri nets are suitable for answering the different questions to a system.

The class of Petri nets, that has been used for qualitative analysis are the non-stochastic and even non-timed Petri nets. Their definition and analysis, which is the calculation of place and transition invariants and proves of structural properties are carried out in [Mur89, Rei85, Sta90, DA92]. Petri nets have been successfully applied to software analysis [BDF$^+$94], work flow processes and manufacturing systems [Sch92].

Petri net models of real applications tend to be extremely large and are unconvenient to handle. To facilitate the model construction concepts like colours for tokens have been introduced [Jen92]. The tokens can be of different type and undergo specific treatment according to their type. Similarly, in predicate-transition nets [GL81] the tokens are expressions that are manipulated by the firing of transitions. All high-level Petri net formalisms [JR91] aim at more compactly representing an originally confusing net. A specialized class of coloured Petri nets has recently also been used for the quantitative analysis [Zim97].

Continuous state Petri nets for the qualitative analysis of technical systems have been introduced by David and Alla [DA93]. They are derived by splitting tokens into an infinite number of small particles. Just recently active research has started in the field of *hybrid Petri nets*. They are applied to railway modelling [DS98] and to hybrid control [DK98, Fla98]. They have been included in a high-level formalism [GU98] and augmented with differential equations [Dra98].

Most of these approaches either combine discrete Petri nets with differential equations, or discretize the continuous state space at some point, which always results in tremendous problems with complexity.

It is not yet possible to say, which of the solutions are the most promising, since they all are in early phases of development.

In the Petri net classes intended for quantitative analysis concepts for the duration of actions are required. So time has been associated with the transitions, that is the firing is assumed to happen after a delay. This formalisms has already been used for performance evaluation [RH80]. Soon afterwards, the delay has been taken to be of exponentially distributed duration. Among the first to publish *stochastic Petri nets* was Michael Molloy [Mol82].

A state of a stochastic Petri net (SPN) is defined by the number of tokens in each place and a state transition is caused by the firing of a transition. In an SPN all transitions fire after an exponentially distributed delay. Hence, the state space gives rise to a continuous timed Markov chain (CTMC), which also underlies a queueing system [Kle75]. So both, queueing systems and stochastic Petri nets, are equivalent in their underlying stochastic process, but Petri nets allow for concurrency and synchronization and are therefore more powerful.

However, pure logical synchronization can not be modelled in SPNs. For that sake *immediate* transitions have been introduced in [ACB84], which fire without delay and are therefore useful only for representing logical dependencies. This formalism has been named *Generalized stochastic Petri nets* (GSPNs). It is the basis for many Petri net classes that have been defined later. A formalism very close to GSPNs but more powerful are *stochastic reward nets* (SRN) [CBPFC$^+$93], that include inhibitor arcs, marking-dependent arc cardinalities and reward structures. Stochastic activity networks (SANs) [CFJ$^+$91] are another Petri net like modelling formalism. In addition to places and transitions they have *gates* to hold input and output conditions and functions of transitions. Both, SRNs and SANs have additional structures for the definition of rewards and are therefore especially suitable for reward modelling. In this thesis the focus is on the modelling formalism rather than on the reward structures. Therefore the basic Petri net class are GSPNs. The above mentioned stochastic Petri net classes are very powerful modelling formalisms, but still their applicability is restricted in some fields. The shortcomings, that have been addressed over the past twelve years are two-fold. One is the limitation to the exponential distribution in the duration of all actions, which is often not appropriate, and the other is the problem of complexity. What is meant here, is not in first place the complexity of the Petri net graph, although this certainly also is an issue [Zim97], but the size of the state space, that often renders analytical solution impossible because of insufficient memory. Very recently, the purely discrete state space has been recognized as a limitation, that should be no longer accepted. In the following all three areas of extensions of GSPNs will be discussed.

To extend GSPNs for the modelling of actions that are not exponentially distributed, at first deterministic firing times have been connected to timed transitions [AC87], leading to a formalism called *Deterministic and stochastic Petri nets* (DSPNs). Phase-type distributions have been introduced in [Ajm90a]. They can be represented by a net of exponentially distributed activities [Neu94] and replace many different kinds of distributions, but add to the state space. Later, not only deterministic firing times, but arbitrary probability distributions have been considered [GL94]. The transitions will be simply called *general transitions* or *deterministic transitions* if their firing time is either of a general distribution or deterministic. Of course, deterministic transitions are a special kind of general transitions.

If no more than one general transition is enabled at once, the underlying stochastic process is a Markov regenerative process and the Petri net class is called *Markov regenerative stochastic Petri nets* (MRSPNs) [CKT94]. An MRSPN can be solved by applying Markov renewal theory [Cin75] with embedding a Markov chain at appropriately chosen points in time.

The firing probability of the general transition depends on the time it has been enabled already and hence the stochastic process is not Markovian while a general transition is enabled. So the question of *firing policies* arises. A firing policy specifies, whether a general transition is reset with every state change (race with resampling), when being disabled (race with enabling memory) or whether the already passed activation time is kept in memory (race with age memory) [TBJ+95, TB95]. An overview of the Petri net classes and their properties and analysis is given in [Ajm90b]. Another solution method for MRSPNs (again with the restriction, that only one general transition can be enabled in every marking) is the method of supplementary variables [Cox55]. A continuous variable is included in the state description and so the stochastic process becomes Markovian again, but with a mixed continuous and discrete state space [GL94]. Thanks to the well-known and regular behaviour of the supplementary variable (it represents time and therefore always increases linearly), the resulting differential equations are only ordinary and of first order [HG97] and can be solved with the uniformization technique [Gra77, GM84].

If more than one general transition is enabled in a state, the stochastic process and the underlying differential equations can still be formulated in different ways [Ger94, LS96], but the equations have yet been solved only for models with two concurrent general transitions.

A solution method for DSPNs with many deterministic and exponential transitions has been proposed in [Zij97]. The model is solved in discrete time [Mol85] by replacing all firing time distributions by their discrete equivalents and then mapping the net on a discrete time Markov chain [Cin75]. The problem here is the relation of the firing times of concurrently enabled deterministic transitions which determines the size of the discrete time steps and hence the number of discrete time steps needed for a deterministic firing time. Also in discrete time the resolution of conflicts is a problem.

The second problem of GSPNs, that of course similarly arises in MRSPNs, is the explosion of the state space. This is mainly a problem concerning the numerical analysis. For simulation of Petri nets the state space does not have to be generated completely, so only simulation times increase dramatically, but simulation is still feasible.

An attempt to overcome this problem is made by decomposing a Petri net into different subnets. Decomposition is carried out either on the net level or on the level of the state space. There are models, where the state space can be split in parts that are connected by highly unlikely state transitions. Then the model can be solved using Tensor algebra. The state space is represented by the Kronecker sum of the sub spaces and the model can be solved in a fixed point iteration over the subspaces [CT93, MT95]. This method gives exact solutions but is applicable only for large Markovian systems that are not too strongly connected.

For performability models upper and lower bounds have been computed by truncating the state space [Hav93]. An upper bound of the performance is obtained by dropping all the system's down-states and solving the rest of the model. Similarly a lower bound is calculated by dropping the up-states.

Queueing models with more general arrival and service time distributions have been solved on the level of the model [Kue79, Whi83, Hav95]. The streams of tokens along single arcs is regarded, giving an approximate solution of the model without the construction of the reachability graph.

If only one place in the Petri net is responsible for the state space explosion because it is unbounded, spectral expansion or the matrix-geometric method can be applied [HO97, MC95].

Furthermore, different ways of hierarchically structuring PN models have been studied in [Buc91, Buc92]. As well as the coloured Petri nets [Jen92] they show the structure of a model more clearly, so that independent subparts can be found and analyzed separately. However, automation of the decomposition process is a problem.

The limitation to a purely discrete state space has been tried to overcome with the introduction of *fluid places* in GSPNs [TK93, HKNT98], at the same time as the first continuous approach in the non-stochastic Petri nets has been published. The fluid places contain a continuous amount of fluid and they are filled and emptied at deterministic rate. This corresponds to the first order fluid models in chapter 3. Second order fluid Petri nets have not been found in the literature prior to the word reported in this thesis. The numerical solution of *fluid stochastic Petri nets* (FSPNs) is not at all straight forward because it involves the solution of partial differential equations. Therefore, simulations seems advantageous. Some simulation issues have been discussed in [CNT99] and implementations have been presented in [NM95, Mac97].

1.3 Contribution of this Thesis

It has been seen in the previous section, that fluid Petri nets still are in an early stage of development. They are meant to be used for performance and dependability analysis of systems containing continuous components such as time, liquid, temperatures, or others [WH97]. But they can also be used for measuring or accumulating reward [Hor96] or for approximation purpose in situations where items are processed following a non-exponential distribution [Kob74a, Kob74b].

In queueing theory fluid models have been studied extensively (chapter 3 will give an introduction). The aim was, to be able to handle other than exponentially distributed arrival and service times. A continuous state model is obtained from a discrete queueing model by applying an approximation method. Many different kinds of approximations have been used [Kle76]. The most important ones are the heavy traffic, the fluid flow and the diffusion approximation.

In the heavy traffic approximation the waiting time for a queueing system with generally distributed arrivals and service times is shown to be exponentially distributed in situation of heavy traffic. Fluid flow approximation is a first order approximation, where arrival and service times are replaced by their expectations, leading to a deterministic flow into and out of the fluid place. Finally, diffusion approximation is a second order approximation, which means that roughly spoken, arrival and service process are replaced by their first two moments. It has turned out, that diffusion approximation gives the best results, especially in situations of congestion. This seems natural, since in the derivation of the diffusion approximation the central limit theorem is used.

The aim of this thesis is to further develop fluid stochastic Petri nets and [HKNT98] is taken as a starting point. On the level of the Petri net two major extensions will be made.

The first is, that the flow of fluid into a fluid place and out of a fluid place will be assumed to vary randomly instead of being deterministic. This is equivalent to using the diffusion, instead of the fluid flow approximation. So, what has proven to be a powerful approach in queueing systems will be transferred to Petri nets.

The second fundamental extension will be the introduction of jump transitions for the fluid part of a Petri net. On the level of the specification a similar construct has been used already in [CNT99]. In the modelling of hybrid systems situations arise, where a continuous component is reset to a certain value instantaneously. Jump transitions will make it possible to have immediate transitions being connected to fluid places under certain restrictions. The example used is that of unfinished work in a queueing system, where upon each arrival of a customer, its service time is added at once to the total amount of unfinished work (or virtual waiting time) in the system.

For numerical reasons the fluid places always have a limited capacity and it can happen, that the amount of fluid filled in instantaneously will exceed the limitation. This situation can be handled in two different ways: either the draining in is stopped and not carried out at all, or the amount filled in is reduced, so that the bound is respected. Both boundary treatments will be regarded and implemented. The same holds for emptying a fluid place instantaneously.

The diffusion equation for queueing systems can be solved analytically in many cases which is no longer the case for second order FSPNs. They always have to be solved numerically. There are few special cases, where an analytical solution still exists. The extended FSPN formalism has been called second order FSPNs because the diffusion process is described by a system of second order partial differential equations. In this thesis, the Petri net class will be formally defined. The Petri net will be mapped on a stochastic process, that is a Markov process with mixed continuous and discrete state space. The differential equations will be derived and a numerical solution method for the transient analysis of second order FSPNs with one and two fluid places is developed and implemented.

The contribution of this thesis is to make available a formalism and transient solution components for performance and dependability modelling of hybrid systems. A system may be hybrid by nature or through approximation of model parts or inclusion of continuous reward measures.

1.4 Organization

This thesis is divided in three parts. The first part covers the background and includes chapters one through three, the second includes the theory in chapters four and five and the third contains application and evaluation of the theory in chapters six through eight.

In chapter two the basic definitions and methods for the analysis of stochastic Petri nets will be introduced. This is done to describe the formal foundation on which the fluid stochastic Petri nets are set up.

Chapter three is devoted to theory and application of fluid queueing systems that can be seen as the second major root of the work reported in this thesis.

In the second part chapter four is the analogue to chapter two, but for fluid Petri nets and in many ways extends the definitions from chapter two.

In chapter five the differential equations, that describe the dynamics of an FSPN, are derived. Equations are derived for the transient as well as for the stationary analysis and for models with only one and models with many fluid places.

The numerical methods needed to solve the differential equations will be discussed in chapter six.

Four different applications will be presented in chapter seven. First a queueing model with non-exponentially distributed service, where the approximation quality of the diffusion approximation will be discussed. Second is a performability example where accumulated reward is modelled with a fluid place. The third example is a hybrid batch reactor, that needs two fluid places, one for the reactor content and one for its temperature. Finally the last example is again a problem from queueing theory, the computation of unfinished work in a queueing system. The fluid place holds the unfinished work and upon arrival of a customer to the system, the fluid level undergoes a jump up by the height of the customers service time. Comparisons with known results will be carried out.

The last chapter concludes this thesis and discusses topics for further research.

Chapter 2

Stochastic Petri nets

Generalized stochastic Petri nets are very popular in the quantitative analysis of technical systems. They have been used as basic formalism for many more specialized model classes, among them the FSPNs as defined in [HKNT98].

As a consequence, they will also be the basis for the FSPN class defined here. In order to be able to refer to the exact formalism and it's analysis (again simulation will not be regarded) both are briefly discussed in this chapter.

In the first section the components of GSPNs will be defined and explained and an example will be given. The example will be used throughout the thesis and will later be modelled with an FSPN. In the second part of this chapter the probabilistic analysis of GSPNs will be described. The SPN has to be mapped on a stochastic process. Equations for the transient as well as the stationary analysis of the stochastic process have to be derived and solved in either analytical or numerical fashion. The steps will be demonstrated in the example.

2.1 Definition of Stochastic Petri nets

There are many different ways of defining a stochastic Petri net. The most common notation, following [ABCD95] and [CBPFC$^+$93] will be adopted here in many parts. But still, some names and functionalities will be chosen differently, so that the definition of FSPNs in chapter 4 will give a homogeneous framework.

Definition 2.1 (Stochastic Petri net) A stochastic Petri net is a bipartite directed graph that is defined as a 9-tuple:

$$SPN = (\mathcal{P}, \mathcal{T}, \mathcal{I}, \mathcal{O}, \mathcal{H}, s_0, g, \lambda, w)$$

where

- $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{P}|}\}$ is a set of *places* that can hold a discrete number of *tokens*. A *marking* is defined by the number of tokens in each place $p \in \mathbb{N}^{|\mathcal{P}|}$. The number of tokens in a place is denoted as $\#(P)$.

- $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\} = \mathcal{T}_E \cup \mathcal{T}_I$ is a set of *transitions* that can be either *timed* or *immediate*. All timed transitions have exponentially distributed firing times. No other firing times will be regarded in this thesis.

- $\mathcal{I} \subseteq \mathcal{P} \times \mathcal{T}$ is the set of all *input arcs* from a place to a transition.

- $\mathcal{O} \subseteq \mathcal{T} \times \mathcal{P}$ is the set of all *output arcs* from a transition to a place. As well as the input arcs they have an arrow head at their destination.

- $\mathcal{H} \subseteq \mathcal{P} \times \mathcal{T}$ is the set of all *inhibitor arcs* from a place to a transition. They have a little circle instead of the arrow head at their destination.

  Associated with input, output and inhibitor arcs are multiplicity functions $I : \mathbb{N}^{|\mathcal{P}|} \to \mathbb{N}, O : \mathbb{N}^{|\mathcal{P}|} \to \mathbb{N}$ and $H : \mathbb{N}^{|\mathcal{P}|} \to \mathbb{N}$ that assign a cardinality to each arc in every marking. A transition is enabled only if each input place holds at least the number of tokens corresponding to the outgoing input arc's multiplicity in that marking. An inhibitor arc disables a transition if the required number of tokens is in its input place.

- s_0 is the *initial marking*. The set of all markings that are reachable from the initial marking is called the *reachability set*

$$\mathcal{S} = \{s | s_0 \xrightarrow{*} s\}$$

- $g_t : \mathbb{N}^{|\mathcal{P}|} \to \mathbb{B}$ is the guard of transition t. A transition is enabled only if the guard evaluates to *true*.

- $\lambda_t : \mathbb{N}^{|\mathcal{P}|} \to \mathbb{R} \cup \{\infty\}$ is the rate of the exponential distribution of the firing time of transition t. Immediate transitions have rate infinity $\lambda_t(s) = \infty$ and their firing time is zero.

- $w_t : \mathbb{N}^{|\mathcal{P}|} \to \mathbb{R}$ is the weight function of an immediate transition that is enabled in a vanishing marking. The firing probability of each immediate transition t is:

$$\frac{w_t(s)}{\sum\limits_{t \text{ enabled in } s} w_t(s)}$$

■

With these elements a stochastic Petri net model can be constructed, although not all of them need to be used. If arcs are not labelled with their multiplicity, it is assumed to be one. Similarly, if immediate transitions are not labelled with weights, they are assumed to be all equal. A guard is a short form for a sub model, that also does not need to be there.

Guards, transitions rates and transition weights are functions of the marking. That means, they can take on different values for the same transition, depending on the global state of the net.

The extensions to the formalism are in associating different firing time distributions with the transitions, and so splitting the set of transitions further in subsets, and then firing policies need to be specified. Or the set of places is split into the discrete and the continuous, as it will be done here. Another way of extending the SPN formalism is, to introduce additional elements between places and transitions, like the gates in stochastic activity networks, that can hold complex enabling conditions. A different class of extensions is made by introducing different levels of abstraction.

Example

As an example a simple queueing system is used, which will later on be extended and transformed into an FSPN. The server is subject to break-downs and repair, so the system constitutes a performability model. The performance is measured by the throughput of the server and the dependability in the fraction of time, the server is operational.

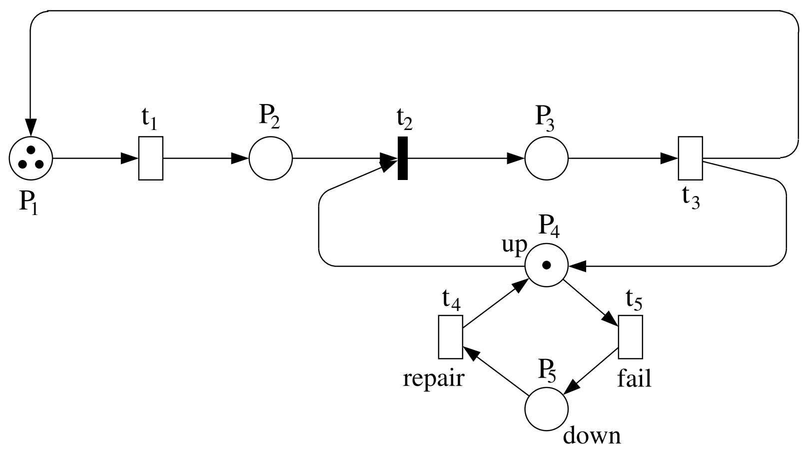

Figure 2.1: SPN of a queueing system

The system has a capacity of three, which is indicated by the tokens in P_1. The places P_2 (waiting room) and P_3 (service station) represent locations, where customers can be situated, P_5 and P_6 indicate the state of the server, whereas the transitions represent the activities in the system. Transition t_1 stands for the arrival of a customer, t_3 for its service, t_4 and t_5 for failure and repair of the server. All these actions are assumed to take an exponentially distributed amount of time. That is, a customer is assumed to arrive after every exponentially distributed inter arrival time and is served for an exponentially distributed service time. Analogously it is assumed, that the server fails after exponentially distributed operational periods and the repair also takes an exponentially distributed time.

The only immediate transition, t_2, in cooperation with place P_4 makes sure, that only one customer at a time can be served, so its function is purely logical.

A state of the system is defined by the status of the server (up or down), and the number of customers in the waiting room and the service facility. It is represented as the *marking* of the SPN, the distribution of the tokens in the places of the net.

2.2 Analysis of Stochastic Petri Nets

In order to analyze a stochastic Petri net the underlying stochastic process must be found. So first the *reachability graph* is built up, a directed graph where the vertices are the reachable states of the Petri net and the edges indicate the transitions between states (for an algorithm see [CBPFC+93]). They usually are labelled with the names of the transitions whose firing causes the change of state.

There are two kinds of states: the *vanishing* states in which no time is spent because they precede the firing of an immediate transition and *tangible* states, in which exponentially distributed transitions are enabled and time is spent. The *reduced reachability graph* (see figure 2.3) is obtained by eliminating all the vanishing markings [CBPFC+93]. The reachability graph is a graphical representation of a stochastic process. For the class of Petri nets discussed so far this is a Markov process and the differential equations describing its temporal behaviour are well known.

The transition rates from one state to another, which are the rates of the exponentially distributed firing times associated with the timed transitions, are the entries of the *generator matrix* $\mathbf{Q}$. The diagonal elements q_{ii} are the negative sum of all other entries in a row $q_{ij}, i \neq j$, so that all rows sum to zero.

$$\mathbf{Q} = (q_{ij}) \quad \text{and} \quad q_{ij} = \begin{cases} \lambda_t & \text{if} \quad i \neq j, \quad i,j \in \mathcal{S}, i \xrightarrow{t} j \\ -\sum\limits_{j,j\neq i} q_{ij} & \text{if} \quad i = j. \end{cases} \tag{2.1}$$

The state probabilities are the entries of the vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_S)$. The transient behaviour of the Markov process is described by the Kolmogorov forward equation [Fel68a], a system of ordinary differential equations (ODEs)

$$\frac{d}{dt}\boldsymbol{\pi}(t) = \boldsymbol{\pi}(t)\mathbf{Q}. \tag{2.2}$$

It corresponds to the Kolmogorov forward equations for a continuous time Markov chain. It can be solved with standard ODE solvers such as Runge-Kutta [Sto94], but also the analytical solution is known to be

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0)\ \mathrm{e}^{\mathbf{Q}t}. \tag{2.3}$$

The matrix exponential is defined as

$$\mathrm{e}^{\mathbf{Q}t} = \mathbf{I} + \sum_{n=1}^{\infty} \frac{\mathbf{Q}^n t^n}{n!} \tag{2.4}$$

but its computation is numerically very unattractive because the negative as well as positive entries might introduce large round-off errors. Therefore, another method for computing (2.3) has become popular, the *uniformization* technique [Jen53]. The idea is to map the original CTMC onto a DTMC and a Poisson process. The Poisson process determines the length of the epochs between the events of the DTMC. A DTMC is specified by its transition matrix $\mathbf{P}$, that is computed from the generator matrix $\mathbf{Q}$ as

$$\mathbf{P} = \frac{1}{q}\mathbf{Q} + \mathbf{I} \qquad \text{and} \quad q \geq \max\{\mathrm{abs}(q_{ii}) | i \in \mathcal{S}\}. \tag{2.5}$$

This definition assures, that $\mathbf{P}$ is a stochastic matrix. All entries are positive and the sum of each row equals one. The following transformations on (2.3) can be carried out

$$\begin{aligned} \boldsymbol{\pi}(t) &= \boldsymbol{\pi}(0)\ \mathrm{e}^{\mathbf{Q}t} = \boldsymbol{\pi}(0)\ \mathrm{e}^{(\mathbf{P}-\mathbf{I})qt} && (2.6)\\ &= \boldsymbol{\pi}(0)\ \mathrm{e}^{\mathbf{P}qt} \cdot \mathrm{e}^{-\mathbf{I}qt} && (2.7)\\ &= \boldsymbol{\pi}(0) \sum_{k=0}^{\infty} \mathrm{e}^{-qt} \frac{(qt)^k}{k!} \mathbf{P}^k && (2.8) \end{aligned}$$

where $\mathrm{e}^{-qt}\frac{(qt)^k}{k!}$ is the Poisson probability of k events happening in time t, if events take place at a rate of λ per time unit [Kar66].

Since $\boldsymbol{\pi}(0)\mathbf{P}^k = \boldsymbol{\pi}(k) = \boldsymbol{\pi}(k-1)\mathbf{P}$ (2.8) can be formulated as

$$\boldsymbol{\pi}(t) = \sum_{k=0}^{\infty} \mathrm{e}^{-qt} \frac{(qt)^k}{k!} \boldsymbol{\pi}(k) \tag{2.9}$$

which directly lends itself to being computed iteratively in k. For the numerical computation the infinite sum needs to be truncated and the truncation point is determined through an error bound for the desired accuracy. Problems like a good choice of q, or a good method for the computation of the truncation point of the infinite sum have been studied intensively over the last years [GM84, vMS97]. Stiff Markov chains have been investigated in particular in [MMT94, RT88, BT86].

It has been proven [vD90] that also Markov processes with continuous and discrete state space can be transformed into a DTMC and a Poisson process, but no applicable algorithm has been derived yet. Therefore, despite its valuable properties, uniformization will not be used in the context of this thesis. Instead finite difference methods have been implemented. Uniformization for FSPNs has been mentioned in [HKNT98].

The equation for a system in steady state is obtained by taking the limit as $t \to \infty$ in (2.2), which gives a system of linear equations

$$\mathbf{0} = \boldsymbol{\pi}\mathbf{Q}, \tag{2.10}$$

where $\mathbf{0}$ is the vector with all components equal to zero. Together with the normalization condition

$$\sum_{i \in \mathcal{S}} \pi_i = 1$$

a unique solution is defined.

Example

The reachability graph of the model in figure 2.1 is constructed by listing up all the reachable markings and connecting states, that follow from each other with arcs. The arcs are labelled with the names of the transitions whose firing causes the state change.

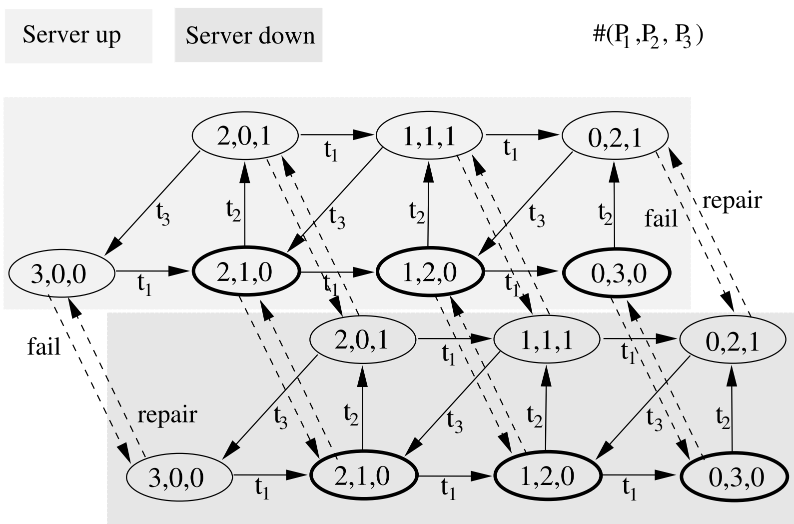

Figure 2.2: Reachability graph of the SPN of a queueing system

Since the system is closed, the reachability set is finite. The state space can be split in two almost identical subsets. The server can fail and be repaired in any state of the system, so the subsets differ only in the status of the server. In order to make the figure better understandable, the status of the server is not part of the labeling of the states in figure

2.2, but is indicated by the shading. There always are transitions from one state in the lower block to the corresponding state in the upper block and vice versa, that are due to the firing of t_4 and t_5 (failure and repair of the server). They are drawn in dashed lines to make the figure easily readable, their meaning is the same, as the other arcs'.

Both blocks have seven states, three of which are vanishing (as indicated by the solid lines surrounding the ovals). In the vanishing states the immediate transition t_2 is enabled.

The vanishing states can be eliminated, giving the reduced reachability graph, that is shown in figure 2.3. In the reduced reachability graph all states are tangible and it is equivalent to a CTMC. As can be seen from the figures, elimination of the vanishing markings has an additional benefit: it reduces the size of the state space considerably.

In figure 2.3 the status of the server is included in the state description, so no shading of blocks is needed. To relate both figures the firing of t_4 and t_5 is still indicated by the dashed arrows. The reduced reachability graph gives rise to a CTMC with eight states, specified

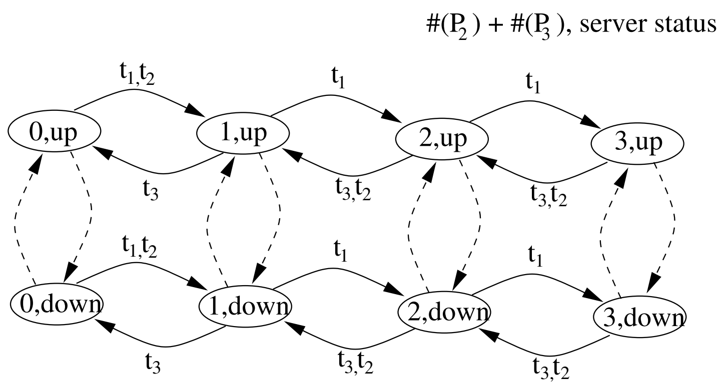

Figure 2.3: Reduced reachability graph of the SPN of a queueing system

by its generator matrix Q. The entries of the generator matrix are the firing rates if the transitions and the entries in the diagonal are the negative sum of the rows, as defined in the first part of this section.

Chapter 3

Fluid Queueing Systems

Over the past forty years a tremendous amount of work has been done in the field of fluid queueing systems, especially concerning diffusion approximation of queues as well as open and closed queueing networks. Queueing systems and Petri nets are closely related in their modeling power. The underlying stochastic processes of both formalisms, that are regarded for the quantitative analysis also are very similar. Therefore, when writing about fluid Petri nets a chapter on fluid queueing systems is mandatory. Various kinds of approximations have been applied to queueing systems [Kle76]. The diffusion approximation is the most widely used and it is applied to Petri nets in this thesis, so it will be the only one regarded for queueing systems and this chapter shall give a non-exhaustive survey of the developments in this area.

Continuous approximations to discrete-valued processes have been used for long [Fel68b], in queueing theory they have been introduced by Newell [New65] for traffic modelling. Besides its application to systems under heavy traffic diffusion approximation has mainly been used to overcome the restriction to Markov processes, i.e. when inter arrival and service times are other than exponentially distributed and in the hope, that here as well as it is often the case, a continuous problem is easier to solve than a discrete one. In heavily loaded queues the central limit theorem holds and the change in the queue length is approximately normally distributed. The parameters of the normal distribution have to be determined from the discrete system, which is not always easy. After solving the diffusion equation the solution has to be transformed back to the discrete valued problem. In the next section this process will be discussed in more detail.

With the more recent developments in telecommunications and high-speed networking attention has been drawn to fluid queueing systems again. Diffusion models have been used to tune parameters in communication protocols and to optimize switches. The second section in this chapter will be devoted to application issues.

3.1 Definition of the Diffusion Approximation for Queueing Systems

The impulse for applying diffusion approximation comes from studies on heavily loaded queueing systems [Kin62] and stochastic car traffic modelling [New65, Gav68]. Overload in shared computer systems has been studied in [Gav71], leading to more specific diffusion models and solutions for backlog and transmission delays. In [New68a, New68b, New68c] it has been shown, that in overload situation with varying rates a transient diffusion model gives more accurate results than deterministic theory or stationary queueing analysis, which were until then the only available methods.

A formal definition of the diffusion process approximating a queueing system for transient as well as stationary analysis has been formulated in [Kob74a, Kob74b]. The diffusion approximation to a single server system is defined as follows.

Let the arrival times of the customers to the system be $0 < t_1 < t_2 < \ldots$ and the departure times $0 < t'_1 < t'_2 < \ldots$. Let $A(t)$ and $D(t)$ be the cumulated numbers of arrivals to the system and departures from the system up to time t. The number of customers in the queue then is

$$N(t) = A(t) - D(t) \tag{3.1}$$

and the change in length of the queue in the interval $[t, t+\Delta]$ is

$$N(t+\Delta) - N(t) = \big((A(t+\Delta) - A(t)) - (D(t+\Delta) - D(t))\big)$$

which is abbreviated as

$$\Delta N(t) = \Delta A(t) - \Delta D(t) \tag{3.2}$$

We assume that the inter arrival times $a_i = t_i - t_{i-1}$ and service times $s_i = t'_i - t'_{i-1}$ are both independent and identically distributed (iid) with mean and variance (μ_a, σ_a^2) and (μ_s, σ_s^2) respectively. Then

$$P\big(A(\Delta) \geq n\big) = P\big(A(t+\Delta) - A(t)\big) = P(\tau_n \leq \Delta)$$

where τ_n is the arrival time of the n-th customer in a time interval of length Δ. Since $\tau_n = t_{\Delta,1} + t_{\Delta,2} + \ldots + t_{\Delta,n}$ is the sum of n iid inter arrival times the central limit theorem holds if Δ is sufficiently large, so that n is not too small [CM65, New65, New71] and τ_n is approximately normally distributed. The same holds for the sum of m iid departure times $\tau'_m = t'_{\Delta,1} + t'_{\Delta,2} + \ldots + t'_{\Delta,n}$. The sum of two normally distributed random variables again is a normally distributed random variable [KT81] (note, that the normal distribution is the only continuous probability distribution with this property) and hence the change in queue length $N(t+\Delta) - N(t) = \tau_n - \tau'_m$ is approximately normally distributed with

$$E\big(\Delta N(t)\big) = \left(\frac{1}{\mu_a} - \frac{1}{\mu_s}\right)\Delta$$

and

$$Var\big(\Delta N(t)\big) = \left(\frac{c_a}{\mu_a} + \frac{c_s}{\mu_s}\right)\Delta$$

where

$$c_a = \frac{\sigma_a^2}{\mu_a^2} \qquad \text{and} \qquad c_s = \frac{\sigma_s^2}{\mu_s^2}$$

are the squared coefficients of variation of the arrival and the service process.

The convergence of Markov, semi-Markov and switching processes [Ani95], that can all be used for describing queueing systems mathematically, to diffusion processes under different load is proven in [Ani95, Ani96].

For the differential equations the infinitesimal mean and variance, which are derivatives with respect to time of the conditional mean and variance (see [Kle76]) are required. They evaluate for all time homogenous arrival and departure processes to

$$E\left(\frac{d}{d\,t}N(t)\right) = \frac{1}{\mu_a} - \frac{1}{\mu_s} \qquad \text{and} \qquad Var\left(\frac{d}{d\,t}N(t)\right) = \frac{c_a}{\mu_a} + \frac{c_s}{\mu_s}.$$

The approximate continuous process $Z(t)$ of $N(t)$ is then chosen such that $d\,Z(t) = Z(t+\Delta) - Z(t)$ is normally distributed with mean μdt and variance $\sigma^2 dt$ where

$$\mu = \frac{1}{\mu_a} - \frac{1}{\mu_s} \qquad \text{and} \qquad \sigma^2 = \frac{c_a}{\mu_a} + \frac{c_s}{\mu_s}. \tag{3.3}$$

Please note that μ_a and μ_s are the mean inter arrival and the mean service times, whereas μ is the mean flow rate. Hence, $Z(t)$ is defined by the stochastic differential equation ([CM65] p. 208)

$$d\,Z(t) = \mu\,dt + X(t)(\sigma^2\,dt)^{1/2} \tag{3.4}$$

where $X(t)$ is a white-noise Gaussian process with mean zero and unit variance.

The interest in continuous state space random processes comes originally from physics. In 1827 the botanist Robert Brown first observed the movement of a small particle in a fluid environment under the random impact of neighbouring particles. The phenomenon therefore is called *Brownian motion*. Only in the early nineteenth century Einstein and Wiener developed a mathematical theory. Since the Markov property holds for diffusion in fluid environment the Markov processes with continuous state space are called *diffusion processes*.

Diffusion processes can also be seen as the continuous limit of discrete processes, where the discrete steps become infinitely small. The continuous limit of the simple random walk (starting at zero, with one-step transitions only to the closest neighbouring states) is called Wiener process, or Brownian motion process. Equation (3.4) is a stochastic differential equation of a diffusion process. If $\mu = 0$ $Z(t)$ is called the standard Wiener or Brownian motion process mentioned above. For $\mu \neq 0$ $Z(t)$ is called Wiener process with *drift* μ.

Given the initial value z_0 the unrestricted process has the following conditional probability density function

$$p(t, z; z_0) = \frac{1}{\sqrt{(2\pi\sigma^2 t)}} \exp(-\frac{(z - z_0 - \mu t)^2}{2\sigma^2 t})$$

that satisfies the Kolmogorov forward equation

$$\frac{\partial}{\partial t} p(t, z; z_0) = \frac{1}{2}\sigma^2 \frac{\partial^2}{\partial z^2} p(t, z; z_0) - \mu \frac{\partial}{\partial z} p(t, z; z_0) \tag{3.5}$$

where $p(t, z; z_0)$ is the probability of Z taking on a value in the infinitesimal environment of z, given that at time $t = 0$ it was z_0, $\frac{d}{dz} P(Z(t) \leq z(t) | Z(0) = z_0)$.

The Wiener process is a diffusion process with constant, independent increments. A more general process is the continuous limit of the homogeneous random walk with variable transition probabilities. It results in the Kolmogorov forward equation with variable coefficients, the Fokker Planck equation

$$\frac{\partial}{\partial t} p(t, z) + \frac{\partial}{\partial z} \big(p(t, z)\mu(z)\big) - \frac{1}{2}\big(\frac{\partial^2}{\partial z^2} p(t, z)\sigma^2(z)\big) = 0. \tag{3.6}$$

A special case, where $\mu(z) = -z\mu$ is the Ornstein-Uhlenbeck process [KT81] with differential equation

$$\frac{\partial}{\partial t} p(t, z) - \mu \frac{\partial}{\partial z} \big(z\, p(t, z)\big) - \frac{1}{2}\big(\frac{\partial^2}{\partial z^2} p(t, z)\sigma^2(z)\big) = 0 \tag{3.7}$$

which does not have independent increments.

A solution to equation (3.5) with a reflecting barrier $Z(t) \geq 0$ is sought. In [CM65] it has been shown (the analogous proof will be carried out for the fluid Petri net in chapter 5), that the boundary condition obtained is

$$\frac{1}{2}\sigma^2 \frac{\partial}{\partial z} p(t, z; z_0) - \mu p(t, z; z_0) = 0 \quad \text{at} \quad z = 0, \quad \forall t. \tag{3.8}$$

The solution is obtained by the method of images [Fel68b, Som49, CM65] as

$$p(t, z; z_0) = \frac{1}{\sqrt{\sigma^2 2\pi t}} \exp\left(-\frac{(x - x_0 - \mu t)^2}{2\sigma^2 t}\right). \tag{3.9}$$

The stationary solution is obtained as $t \to \infty$. For $\mu \geq 0$ $p(t, z; z_0)$ approaches zero for all z, but for $\mu < 0$ the equilibrium distribution is

$$\begin{aligned} p(z) &= 2\frac{|\mu|}{\sigma^2}\exp(-2|\mu|\frac{z}{\sigma^2}) & (3.10) \\ &= 2\frac{1-\rho}{c_s + c_a\rho}\exp(-2\frac{(1-\rho)z}{c_s + c_a\rho}) \quad z \geq 0 & (3.11) \end{aligned}$$

and $\rho = \frac{\mu_s}{\mu_a}$.

From the solution to the continuous process an approximation to the queue size distribution is sought.

The probability vector $\hat{p}(n)$ is defined by integrating $p(z)$ over the unit interval $n \leq z < n+1$:

$$\hat{p}(n) = \int_n^{n+1} p(z)\, dz = (1-\hat{\rho})\hat{\rho}^n \qquad n = 0, 1, 2, \ldots \tag{3.12}$$

with

$$\hat{\rho} = \exp(-2\frac{1-\rho}{c_s + c_a\rho}) \tag{3.13}$$

Eqn. (3.12) looks surprisingly similar to the queue size distribution of an M/M/1 queueing system [Kle75]

$$p(n) = (1-\rho)\rho^n.$$

For $c_a = c_s = 1$ and ρ close to one this approximation is very good [Kob74a]. The utilization factor is determined as $1 - \hat{p}(0) = \hat{\rho}$, but it is known to be $\rho = \frac{\mu_s}{\mu_a}$. So, apparently the approximation for the empty queue is bad. Therefore the queue size distribution is modified to

$$\hat{p}(n) = \begin{cases} 1-\rho & n = 0 \\ \rho(1-\hat{\rho})\hat{\rho}^{n+1} & n \geq 1 \end{cases} \tag{3.14}$$

which is still a geometric distribution for $n \geq 1$.

In [Kob74a] a vector valued diffusion process has been defined for open and closed queueing networks and results analogous to the ones presented above have been derived. The accuracy of the diffusion approximation has been studied in [RK74] where analytical solutions for the M/G/1 and the GI/M/1 queue have been derived and compared with their diffusion approximation. It has been shown, that the approximation gets worse, the larger the deviation in the distribution's squared coefficient of variation (scv) from one is. For distributions with a scv close to one, the approximation is very good. But also the definition of σ^2, μ and the boundary condition is important [Gel75].

In [GP76] a slightly different approach has been chosen for a queueing network. The single queues are treated as being independent as they are in an open network with Poisson arrivals and exponential service times. This leads to different parameters of the diffusion process which the authors state to give more exact results. Transient formulas for the queue length distribution are formulated in [Kob74b].

For the analysis of queueing systems with time-dependent parameters a similar approach as in [Kob74b] has been used in [Dud86]. The time-dependent parameters are discretized and the queues in a network are handled as if they were independent of each other, only connected by input and output rates, that are determined iteratively. Constant output rates have been regarded in [CS95].

The models in [Dud86] are solved subject to *elementary return boundaries*, which have been proposed in [Gel75, Gel79], in order to make the diffusion equation more suitable for light traffic situations. The elementary return boundaries are not reflecting immediately, but it is assumed, that a particle reaching a boundary will stay there for an exponentially distributed time period (other distributions have been treated in [Gel79]) and then jump back to a randomly chosen place in the middle of the regarded domain.

Let $m_1(t)$ and $m_2(t)$ be the probability masses that are concentrated at the lower and the upper bound (assuming there are two boundaries) and $1/\lambda_1$ and $1/\lambda_2$ the expected holding times at the boundaries. The probability density of the points where the particle starts again after having jumped to the interior from the lower or upper bound are $p_1(z)$ and $p_2(z)$, respectively.

The differential equations of this process are

$$\frac{\partial}{\partial t}p(z,t) + \frac{\partial}{\partial z}(\mu(z,t)p(z,t)) - \frac{1}{2}\frac{\partial^2}{\partial z^2}(\sigma^2(z,t)p(z,t)) = \lambda_1 m_1(t)p_1(z) + \lambda_2 m_2(t)p_2(z) \tag{3.15}$$

$$\frac{d}{d\,t}m_1(z) = -\mu(0,t)p(0,t) + \frac{1}{2}\frac{\partial}{\partial z}(\sigma^2(0,t)p(0,t)) - \lambda_1 m_1(t) \tag{3.16}$$

$$\frac{d}{d\,t}m_2(z) = -\mu(z_{max},t)p(z_{max},t) + \frac{1}{2}\frac{\partial}{\partial z}(\sigma^2(z_{max},t)p(z_{max},t)) - \lambda_2 m_2(t) \tag{3.17}$$

where z_{max} denotes the upper bound. It is clear, that the equations (3.5) and (3.8) are included as special cases and although the elementary return boundaries have proven to be suitable for heavy as well as light traffic, they have not been picked up in any later investigations. The reason is most likely, that they are more complex and more parameters have to be specified, and they might be hard to obtain. In this thesis also only the reflecting barrier will be used.

Certain queueing systems have been studied in detail, such as M/G/m [Kim83], GI/G/1 [Hey75] and G/G/k [HF77]. The arrival process has been taken as an Ornstein-Uhlenbeck process [Sim91] and time-dependent behaviour has been studied [Dud86] also for queues with constant output rate [CS95].

The diffusion approximation has been applied to many specific problems, such as fork and join queueing models [Kne91] and two parallel heavily loaded queues [Mor93], or multi-server networks [HF78]. A mixed (open and closed) multi-class (with two types of customers) queueing network has been studied in [Ngu95]. It turned out, that the diffusion approximation gave a reflected Brownian motion on a limited interval for the multi-class queue, which was not the case in the original Markovian model. This is a step towards the diffusion approximation of open multi-class networks, which is still an open problem.

3.2 Applications of Fluid Queueing Systems in Telecommunication

For the performance analysis of the components of computer and telecommunication systems such as switches and communication protocols stochastic modelling is essential.

Queueing models are the most commonly used for switches that are supplied by incoming data lines and that feed outgoing data lines as well as for buffers in a communication protocol where messages are stored. The queueing models allow for the calculation of overflow probabilities in a buffer with a limited capacity, expected waiting times, expected loss rates and others.

In all these models certain assumptions are made concerning the nature of the incoming and outgoing traffic and the processing behaviour of the switch or protocol that often are highly unrealistic and give strongly biased results [Kob74a, RK74].

Diffusion approximation is frequently used for these queueing models and some of the investigations are reported in the present section.

The reasons for the continuous approximation again are two-fold. Either the regarded buffer is almost always filled and a continuous approximation seems appropriate [Gav71] or the distributions of the incoming and outgoing streams can not be handled analytically and therefore the system lends itself for approximation.

Traditionally, Markov models have been used for the performance analysis of communication protocols. Diffusion approximation has been applied to the ALOHA [Kle76] protocol allowing for a more precise modelling of the input data streams as an Ornstein-Uhlenbeck process [CM65, KOH77]. Both the slotted and the unslotted ALOHA have been modelled in a diffusion model and solved and it has been shown that an easy separation of performance parameters is possible permitting an illustrative and accurate analysis. Figure 3.1 shows a diffusion model of an ALOHA system where some of the U^n users try to use the channel and some of the R^n retry to do so.

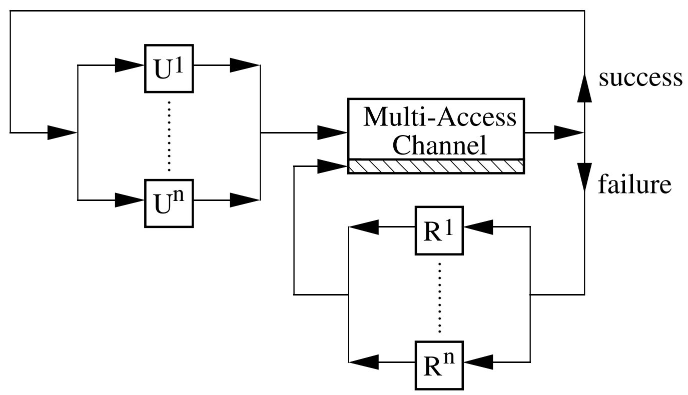

Figure 3.1: Diffusion model for an ALOHA system

Similarly convincing results have been obtained for a token ring system, where each queue in a multi queueing system has been approximated continuously and so the mean waiting time could be calculated for each queue separately [KT86].

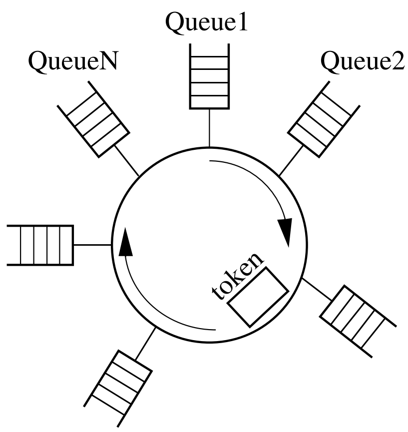

Figure 3.2: Queueing model of a token ring system

A diffusion model for a system with N independent sources that send data on C channels, where the transmission of one packet can be interrupted by the attempt of the next packet to access that channel has been studied in [GL79]. The model is depicted in figure 3.2 and diffusion approximation is applied to all the hatched modules.

However, the ALOHA protocol is no longer in the center of current research and practical use. The token-ring protocol is still used in wireless communication. Today ATM technology demands for new performance models [Onv94] and also the traffic in modern communication systems has changed. For many years mainly data was transmitted and the traffic was assumed to behave like a Markov-modulated Poisson Process (MMPP), which is a doubly stochastic Poisson process, where the rate of the Poisson process itself is

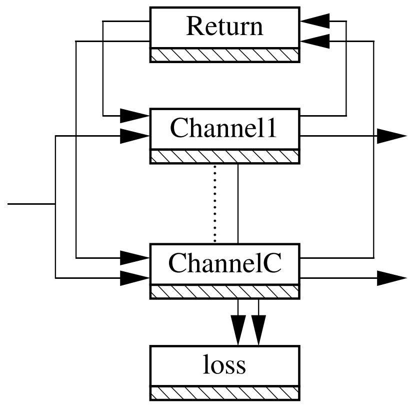

Figure 3.3: Diffusion model of a system with message interference

determined by a continuous time Markov chain. A special case is a Markov chain with only two states and the resulting stream is that of an on-off source. There have been extensive studies about how to match the parameters to get the best fit to the traffic behaviour [HL86].

Other traffic models are that of self-similar traffic [LTWW94], that has been found to be a good model for traffic with long-range dependence [Cox84] like typical Ethernet LAN traffic.

A very popular performance model for the ATM environment has been the model of N independent on-off sources that feed a buffer, suggested in [AMS82]. Active and silent periods are of exponentially distributed duration. The packets, that fill the buffer, are assumed to flow at constant rate. Hence, it is a *first order* model as are the models used in [Mit88, EM94, Box96] and [Hal72, HS72].

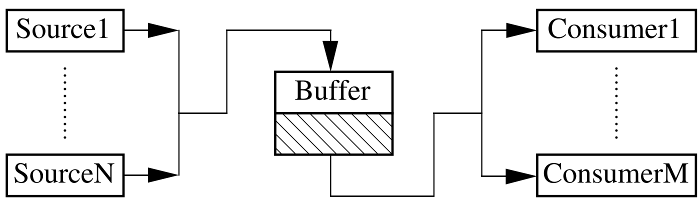

Figure 3.4: Fluid model of a buffer supplied by N sources, feeding M consumers

The model of N independent sources supplying a buffer has been extended by adding M independent consumers fed from the buffer [Mit88], as shown in figure 3.4 and steady-state results for this fluid Markov model have been derived in fairly mathematical analysis with spectral expansion giving all the systems eigenvectors. The model in [AMS82] has been extended further by adding loss-priorities to the input sources (see figure 3.5) [EM94] which is a central element in ATM based Broadband ISDN where raw data as well as coded voice and video cells are transmitted. The input stream is controlled by a Markov modulated Poisson Process (MMPP). From this model guidelines for the design of new multiplexers can be taken. If the active period of one or more input sources is assumed to be of regular variation [Box96], i.e. is not Markovian the heavy tail in the traffic distribution is modelled more correctly.

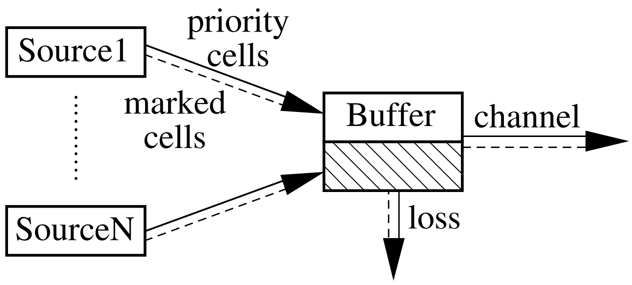

Figure 3.5: Fluid model of a buffer supplied be N sources with loss priorities

Another issue arising in ATM based networks is the call admission control, or the network management. Fluid models have also been used to improve statistical bandwidth allocation strategies for call admission control [GMO96].

All these models differ in the process that regulates the input and output streams, but input and output itself are always modelled as a first order fluid stream, i.e. deterministically. Only recently an extension towards probabilistic flow has been made [KK95, Asm95] with a diffusion model controlled by a MMPP. This second order model is very close to the formalism presented in this thesis. All coefficients are constant, and transient as well as stationary differential equations are derived in [KK95]. The first order (fluid flow) model is studied as a special case. It could be shown, that congestion is underestimated with first order models. The stationary equations can be solved with computing the system's eigenvalues and vectors, but no closed form solution has been found yet.

Chapter 4

Definition and Behaviour of Second Order Fluid Stochastic Petri Nets

In this chapter the formalism of second order FSPNs will be introduced. Second order FSPNs will be abbreviated FSPNs, as are the first order, standard FSPNs. If we want to distinguish between the first order and the second order formalism, the full name will be used for the latter.

With this chapter the second, and central, part of this thesis begins. The topics discussed in earlier chapters, especially chapter 2 and 3 will be used as prerequisites. The GSPNs as discussed in chapter 2 will give the fundamentals in the definition of the FSPNs, whereas the theory of fluid queueing systems is more related to the analysis of FSPNs.

The first section contains the definitions, where the GSPN from chapter 2 will be extended. In the second section the construction of the reachability graph will be presented including the computation of the parameters that define the temporal behaviour of the fluid as well as the discrete model part, and in the third section different cases of fluid flow and jumps will be discussed in more detail. The notion of reachability graph then is needed to explain the way the model syntax is defined, therefore the definition of the reachability graph here forms a separate section and is not part of the analysis.

Interactions of continuous and discrete model parts will be discussed as well as the special cases of constant fluid flow parameters and first order FSPNs.

4.1 The Components of the Second Order FSPN

For the formal definition of an FSPN the definition of an SPN in chapter 2 will be extended. The behaviour of an SPN as described in chapter 2 remains the behaviour of the discrete model part, a continuous part is added and both are interwoven.

Definition 4.1 (Second order fluid stochastic Petri net (FSPN)) An FSPN is a 12-tuple

$$FSPN = (\mathcal{P}_d, \mathcal{P}_c, \mathcal{T}, \mathcal{I}_d, \mathcal{I}_c, \mathcal{O}_d, \mathcal{O}_c, \mathcal{H}, \mathbf{m}_0, g, \lambda, r, w)$$

where

- $\mathcal{P}_d$ is the set of discrete places as defined in chapter 2.
- $\mathcal{P}_c = \{P_{1,c}, \dots, P_{m,c}\}$ is a set of m *fluid places* that can hold a continuous amount of *fluid* rather than discrete tokens and that are drawn by means of two concentric circles. Its initial value is written as a real number in the middle of the circles. For the content of a fluid place the variables $z_i, i = 1, \dots, m$ are used.

  All fluid places have a restricted capacity, that is denoted by the interval $[z_i^{min}, z_i^{max}]$ If only one boundary is defined, it is assumed to be the upper bound and the lower bound is set to zero.

  The complete marking now is defined as composed of two parts, the discrete state and the continuous $\mathbf{m} = (s, \mathbf{z})$ where $s = (\#p_i, i \in \mathcal{P}_d)$ and $\mathbf{z} = (z_k, k \in \mathcal{P}_c)$. The initial marking is $\mathbf{m}_0 = (s_0, \mathbf{z}_0)$.
- $\mathcal{T} = \{t_1, \dots, t_{|\mathcal{T}|}\} = \mathcal{T}_E \cup \mathcal{T}_G \cup \mathcal{T}_I$ is a set of transitions. $\mathcal{T}_E$ and $\mathcal{T}_I$ are as defined in chapter 2, $\mathcal{T}_G$ is a set of transitions with arbitrary firing time distributions. These transitions are drawn as grey rectangles.
- $\mathcal{I}_d \subseteq (\mathcal{P}_d \cup \mathcal{P}_c) \times \mathcal{T}_E \cup \mathcal{T}_I$ is the set of all *discrete input arcs* as defined in chapter 2. They can be used for connecting a fluid place and a transition if the transition is meant to be a *jump transition*. Then it is labelled with the *jump height density function* $b(.)$. An input arc of a transition indicates a jump down, that occurs at the firing rate of the transition, if the transition has an exponentially distributed firing time. A discrete arc can under certain restrictions also connect a fluid place and an immediate transition. This will be specified further in conjunction with the reachability graph in the next section, since the restrictions are due to difficulties in the construction of the reachability graph.
- $\mathcal{I}_c \subseteq \mathcal{P}_c \times \mathcal{T}_E \cup \mathcal{T}_G$ is the set of all *fluid input arcs* leading from a fluid place to a timed transition. They are drawn like pipes. Fluid flows along the arc as long as the fluid place is not empty and the transition is enabled by all its discrete input places, or by a guard, that can also depend on the continuous state.

  If the arc is not labelled with a flow rate function it has to be computed from the firing rates of the transition (see next subsection). If the transition is only connected to fluid places and its firing has no effect on the discrete state of the model, the firing time can have any probability distribution. These are the transitions $t \in \mathcal{T}_G$.

- $\mathcal{O}_d \subseteq (\mathcal{T}_E \cup \mathcal{T}_I) \times (\mathcal{P}_d \cup \mathcal{P}_c)$ is the set of all *discrete output arcs* from a transition to a discrete place. Analogous to the input arcs they can connect a transition and a fluid place. The firing of the transition causes a jump up.

  In order to avoid a jump across the boundaries a parameter needs to be defined, that can take on one of the two values

  - preserve
  - force jump

  where the first means, that a jump, that would go beyond the boundary should not be carried out and the second means, that in any case a jump should happen as far as possible, so the jump height is reduced. Default is the latter.

- $\mathcal{O}_c \subseteq (\mathcal{T}_E \cup \mathcal{T}_G) \times \mathcal{P}_c$ is the set of all *fluid output arcs* analogous to the fluid input arcs.

- $\mathcal{H} \subseteq \mathcal{P}_d \times \mathcal{T}$ is the set of all *inhibitor arcs* as defined in chapter 2.

- $\mathbf{m}_0 = (s_0, \mathbf{z}_0)$ is the *initial marking* consisting of a discrete part as defined in chapter 2 and a vector of initial fluid levels in all fluid places. The set of all markings that are reachable from the initial marking is called the *reachability set*

$$\mathcal{M} = \{\mathbf{m}|\mathbf{m}_0 \xrightarrow{*} \mathbf{m}\} = \mathcal{S} \times \mathcal{Z} = \mathcal{S} \times (\cup_i [z_i^{min}, z_i^{max}]) \subseteq \mathbb{N}^{|\mathcal{P}|} \times \mathbb{R}^m.$$

- $g_t : \mathbb{N}^{|\mathcal{P}|} \times \mathcal{Z} \to \mathbb{B}$ is the guard of transition t that can be a function of the discrete and the continuous state.

- $\lambda_t : \mathbb{N}^{|\mathcal{P}|} \times \mathcal{Z} \to \mathbb{R} \cup \{\infty\}$ is a function of both the continuous and the discrete marking. Apart from that it is defined as in chapter 2.

- $\mathrm{r}_{t,p}, \mathrm{r}_{p,t} : \mathbb{N}^{|\mathcal{P}|} \times [z_p^{min}, z_p^{max}] \to \mathbb{R}^2$, are the flow rate functions associated with the fluid input and output arcs that assign a cardinality in form of a normal distribution to each fluid arc in every marking. A transition is enabled only by the discrete marking and a guard, that can depend on the continuous state as well. Fluid flows along an arc as long as the transitions in its origin is enabled and no upper or lower bound is reached. To preserve the flow rates' independence they can only depend on the fluid level of the fluid place the arc is connected to.

- $w_t : \mathbb{N}^{|\mathcal{P}|} \to \mathbb{R}$ is the weight function of transition t as defined in chapter 2. ■

Note, that a transition with generally distributed firing time can be connected only to a fluid place but not to a discrete one, in which case the arc will not be labelled. Instead, the firing time of the transition should define the fluid flow rate.

With this definition, the elements of an FSPN are introduced. The fluid flow and jump parameters can be used and interpreted in different ways, which will in more detail be discussed together with the reachability graph in section 4.2.

Example

Figure 4.1 shows a model of a queueing system that corresponds to the one used in chapter 2. The server is again subject to failure and repair. In the example in chapter 2 the system was closed, having only three tokens to represent customers. It is not possible in an FSPN to define a closed system, other than by imposing a bound on the fluid place. Neither is it possible to treat customers separately one by one, because they never appear as single items, but rather as a stream of a certain intensity. Therefore the capacity of the fluid place is limited to three. The fluid place rather represents the waiting room, than the actual service station. While transition t_1 is enabled, customers enter the waiting room and while transition t_2 is enabled they are served. The bound on the fluid place, that replaces a submodel allows for a more compact graphical representation. In general, one can say, that the elements an FSPN is constructed from are more complex than the elements of a GSPN, but the resulting model is more compact, than a GSPN would be.

Since this model is meant to be understood as an analogous one to the GSPN in chapter 2, the fluid arcs are not labelled with rates, but they have to be computed from the firing rates of the transitions t_1 and t_2.

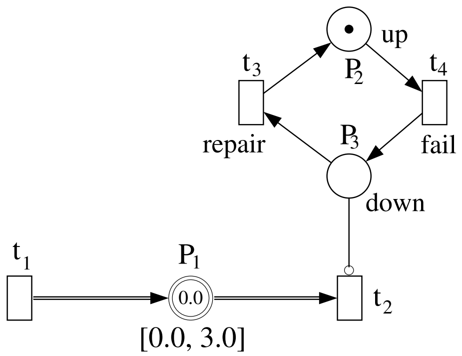

Figure 4.1: FSPN of a queueing system

The arcs are labelled with a rate function, that is either computed from the transitions' firing rates or chosen by the user as shown in the next section.

4.2 The Reachability Graph and the Fluid Parameters of an FSPN

A fluid stochastic Petri net consists of a discrete and a continuous model part. Accordingly the state space is mixed continuous and discrete.

The reachability graph of an FSPN is constructed from vertices that represent the states and edges between the vertices, that represent the discrete state transitions. It mostly looks like the reachability graph of a GSPN. The reachability graph of an FSPN has a vertex for each discrete state. The continuous state space is included completely in every discrete state, represented by a continuous variable for each fluid place.

Although FSPN models can contain transitions with generally distributed firing times the stochastic process underlying an FSPN is a Markov process with mixed continuous and discrete state space. The general transitions do not lead to discrete state transitions but only affect the fluid places. Therefore the general distribution determines the fluid flow rate, but it does not violate the Markov property. The behaviour of the discrete part is represented in the generator matrix $\mathbf{Q}(\mathbf{z})$ that can be directly read from the reachability graph. The discrete transition rates can depend on the continuous state as well. The temporal behaviour of the model in the continuous part of the state space, however, is not directly represented in the reachability graph. Parameters for the fluid flow and the fluid jumps have to be defined separately. There is such a set of parameters for each fluid place. together with the reachability graph they fully specify the underlying Markov process.

We will now first define the reachability graph through an algorithm for its construction. Together with it comes the definition of the generator matrix.

The algorithm for building up the reachability graph as presented in [CBPFC^{+}93] has to be slightly modified. the reachability graph of an FSPN is constructed according to the following algorithm:

$\mathcal{S}_c = \{(s_0, \mathbf{z})\} = \{m_0\}$
$\mathcal{R}_c = \emptyset$
$\mathcal{S}_c^{new} = \{m_0\}$
while $\mathcal{S}_c^{new} \neq \emptyset$ do
    choose a marking i from $\mathcal{S}_c^{new}$
    $\mathcal{S}_c^{new} = \mathcal{S}_c^{new} \setminus \{i\}$
    for each $t \in \sqcup$do
        if $\exists j \in \mathcal{S}$ and $\exists \mathbf{z} \in \mathcal{Z}$ such that $g_t(i, \mathbf{z}) =$ true, and $i \xrightarrow{t} j$
            $m^{new} = (j, \mathbf{z})$
            if $m^{new} \notin \mathcal{S}_c$ then
                $\mathcal{S}_c^{new} = \mathcal{S}_c^{new} \cup \{m^{new}\}$
                $\mathcal{S}_c = \mathcal{S}_c \cup \{m^{new}\}$
            $\mathcal{R}_c = \mathcal{R}_c \cup \{(i, \mathbf{z}) \xrightarrow{t} (j, \mathbf{z})\}$.

and finally $\mathcal{S}_c$ is the complete reachability set and $\mathcal{R}_c$ the complete reachability graph. Attached to all the transitions' names are firing rates for exponential transitions and firing weights for the immediate ones. The model does not spend time in the *vanishing* states, where only immediate transitions are enabled.

The reduced reachability set $\mathcal{S}$ and reachability graph $\mathcal{R}$ are determined by eliminating the vanishing markings.

If the firing of an exponential transition $t \in \mathcal{T}_e$ is followed by one or more more subsequent, or alternative firings of immediate transitions, there are several intermediate vanishing states until the model reaches its next tangible state where an exponential transition is enabled. The reduced reachability graph consist only of tangible markings, so the vanishing markings have to be eliminated. In the implementation both steps are carried out at once.

The following algorithm eliminates the vanishing markings:

$\mathcal{S} = \mathcal{S}_c$
$\mathcal{R} = \mathcal{R}_c$
$\mathcal{S}_c^{new} = \mathcal{R}_c$
while $\mathcal{S}_c^{new} \neq \emptyset$ do
  choose a tuple $(i, \mathbf{z}) \xrightarrow{t} (j, \mathbf{z})$ from $\mathcal{S}_c^{new}$
  if $t \in \mathcal{T}_e$
    $\mathcal{S}_c^{new} = \mathcal{S}_c^{new} \setminus \{(i, \mathbf{z}) \xrightarrow{t} (j, \mathbf{z})\}$
    for each $t' \in \mathcal{T}_i$ with $(j, \mathbf{z}) \xrightarrow{t'} (k, \mathbf{z}), j, k \in \mathcal{S}_c$
      $\mathcal{R} = \mathcal{R} \setminus \{(i, \mathbf{z}) \xrightarrow{t} (j, \mathbf{z}), (j, \mathbf{z}) \xrightarrow{t'} (k, \mathbf{z})\}$
      $\mathcal{R} = \mathcal{R} \cup \{(i, \mathbf{z}) \xrightarrow{t,t'} (k, \mathbf{z})$
      $\mathcal{S} = \mathcal{S} \setminus \{(j, \mathbf{z})\}$
      $\mathcal{S}_c^{new} = \mathcal{S}_{new} \setminus \{(j, \mathbf{z}) \xrightarrow{t'} (k, \mathbf{z})\}.$

Finally $\mathcal{R}$ is the reduced reachability graph and $\mathcal{S}$ the set of all tangible markings. The transition rates from one state to the next are computed as the product of the firing rate of the exponential transition and the weights of the following immediate transitions, $\lambda_{t,t'} = \lambda_t \cdot w_{t'}$. In the following the discrete transition rates will all be taken from the reduced reachability graph, so they are all firing rates of the exponential distribution. Note that there can be exponential transitions in a model, that are not present in the reachability graph. This happens, if they are connected only to fluid places.

The generator matrix $\mathbf{Q}$ of the underlying Markov process is defined as

$$\mathbf{Q}(\mathbf{z}) = (q_{ij}(\mathbf{z}))_{i,j=1,\ldots,|\int|} \quad \text{and}$$

$$q_{ij}(\mathbf{z}) = \begin{cases} \lambda_t(\mathbf{z}) & \text{if} \quad i \neq j, \quad i, j \in \int, \exists \mathbf{z} \in \ddagger \text{ such that } g_t(i, \mathbf{z}) = \text{true, and } i \xrightarrow{t} j \\ -\sum_{j, j \neq i} q_{ij}(\mathbf{z}) & \text{if} \quad i = j. \end{cases} \tag{4.1}$$

We now proceed to the definition of the fluid parameters. A state in the discrete state space is determined by the number of tokens in each place. A new marking is computed by adding and subtracting the number of tokens that are placed and removed upon the firing of a transition. So, for obtaining a new marking calculations have to be carried out with the multiplicities of the arcs as operands. The exact procedure has been omitted here, it can be found in [CBPFC$^+$93]. But the analogous computations have to be carried out for the continuous state space as well. What makes the situation more complicated and impossible to handle explicitly, is that the rates of the fluid arcs, the analogous to the multiplicities of the discrete arcs are random instead of deterministic. Hence, their sum and difference will again be a random variable and no explicit values can be given, except for expected values.

The fluid parameters have to be specified for each discrete state $i \in \mathcal{S}$ and for every fluid place. there are parameters for the fluid flow as well as for the fluid jumps. We first regard the continuous flow. The continuous flow for one fluid place $k \in \mathcal{P}_c$ in a discrete state $s \in \mathcal{S}$ is specified by the *fluid change rate function* $R : \mathcal{P}_c \times \mathcal{S} \to \mathrm{I\!R}^2$, that is a normal distribution, defined through its first two moments, $R(k,s) = (\mu, \sigma^2)$.

There are again two cases to be distinguished for the computation of the fluid change rate. Either the rates along the single fluid arcs have been specified by the user, then

$$R(k,s) = (\mu_{k,s}, \sigma^2_{k,s}) = \left(\sum_{\substack{(t,k)\in\mathcal{O}_c \\ t\in\mathcal{T}}} \mu_r(z) - \sum_{\substack{(k,t)\in\mathcal{I}_c \\ t\in\mathcal{T}}} \mu_r(z), \sum_{\substack{(t,k)\in\mathcal{O}_c \\ (k,t)\in\mathcal{I}_c \\ t\in\mathcal{T}}} \sigma^2_r(z) \right) \tag{4.2}$$

where $(t,k) \in \mathcal{O}_c$ are all the fluid arcs leading from a transition, that is enabled in state s to the fluid place k and $(k,t) \in \mathcal{I}_c$ are the fluid arcs from a fluid place to a transition, respectively. μ_r and σ^2_r are expectation and variance of the rate specified for the arc.

If on the other hand no rate has been assigned to an arc in the FSPN, it is assumed, that the fluid place is meant to replace a discrete place for approximation reasons and the firing rates of output and input transitions are used for the computations of the fluid change rate function.

Let $t_i \in \mathcal{T}_E \cup \mathcal{T}_G$ be the transitions with firing time distribution f_i, connected to the fluid place k by a fluid arc $(t_i, k) \in \mathcal{O}_c$ and let $t'_i \in \mathcal{T}_E \cup \mathcal{T}_G$ be the transitions with firing time distribution g_i, connected to the fluid place k by a fluid arc $(k, t_i) \in \mathcal{I}_c$ and let all be enabled in state $s \in \mathcal{S}$. The distributions f_i and g_i are assumed to have an expectation μ_{f_i} and μ_{g_i} and a variance $\sigma^2_{f_i}$ and $\sigma^2_{g_i}$. The squared coefficient of variation (scv) is defined as $c_{f_i} = \frac{\sigma^2_{f_i}}{(\mu_{f_i})^2}$, and c_{g_i} analogously. Then, $R(k,s)$ is computed like the parameters in fluid queueing systems (see (3.3) in chapter 3) as:

$$R(k,s) = (\mu_{k,s}, \sigma^2_{k,s}) \quad \text{with}$$

$$\mu_{k,s} = \sum_i \frac{1}{\mu_{f_i}} - \sum_i \frac{1}{\mu_{g_i}} \qquad \text{and} \qquad \sigma^2_{k,s} = \sum_i \frac{c_{f_i}}{\mu_{f_i}} + \sum_i \frac{c_{g_i}}{\mu_{g_i}}. \tag{4.3}$$

If f_i and g_i are exponential distributions with parameter λ_i and μ_i, respectively, the scv evaluates to $c = 1$ and the parameters $\mu_{k,s}$ and $\sigma^2_{k,s}$ reduce to

$$\mu_{k,s} = \sum_i \frac{1}{\lambda_i} - \sum_i \frac{1}{\mu_i} \qquad \text{and} \qquad \sigma^2_{k,s} = \sum_i \frac{1}{\lambda_i} + \sum_i \frac{1}{\mu_i}. \tag{4.4}$$

With (4.3) the fluid flow parameters for one discrete state are specified. For all states in the model and for each fluid place they are collected into matrices with entries only in the diagonal elements for the expectation

$$\mathbf{M}_k(z) = \operatorname{diag}\left(\mu_{k,1}(z) \dots , \mu_{k,|\mathcal{S}|}(z)\right) \tag{4.5}$$

and for the variance

$$\mathbf{\Sigma}^2_k(z) = \operatorname{diag}\left(\sigma^2_{k,1}(z), \dots , \sigma^2_{k,|\mathcal{S}|}(z)\right). \tag{4.6}$$

Fluid jumps can result from the firing of an exponential or an immediate transition $t \in \mathcal{T}_E \cup \mathcal{T}_G$, that is connected to a fluid place $k \in \mathcal{P}_C$ by a discrete arc. If t is an exponential transition, the jumps occur in a Poisson process at rate λ_t, each after an exponentially distributed delay in coincidence with the firing of the transition, that can also move tokens from input to output places. On the level of the reachability graph, this means, that jumps due to the firing of an exponential transition can happen concurrently with a discrete state change or exclusively. As opposed to this, jumps due to the firing of an immediate transition may only happen together with a discrete state change. The rate $\lambda_{\bar{t}}$ in the Poisson process, at which the jump occurs, then is determined as $\lambda_{\bar{t}} = \lambda_{\tilde{t}} \cdot w_t$, the rate of the preceding exponential transition times the weight of the immediate transition. It also means, that there has to be an exponential transition firing prior to the immediate transition causing the fluid jump. This will be illustrated in more detail with the examples in the next section.

In summary, a fluid jump occurs in a Poisson process at rate λ with the model being in a certain discrete state, which is left or not upon the jump. There can be several jumps at rate λ_j in one discrete state, or when leaving one discrete state, but each jump rate can be associated with only one discrete state and one fluid place.

In addition to the jump rate a jump height has to be specified. The height is drawn from a probability distribution with density function $b(.)$, so for each jump a density function $b(.)$ has to be defined.

For notational convenience all parameters are collected in matrices $\mathbf{\Lambda}$ for the jump rates and $\mathbf{B}$ for the jump heights. Both matrices have the dimension of the state space and

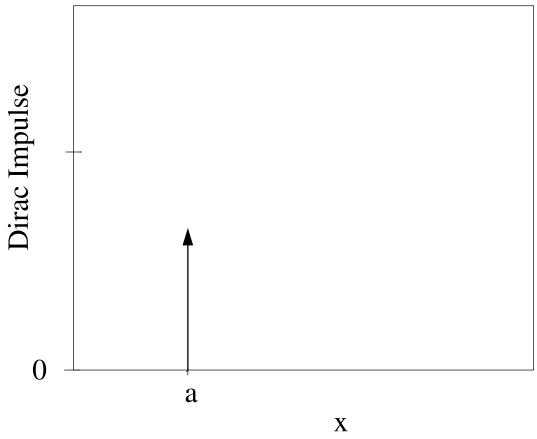

Figure 4.2: Dirac impulse at a

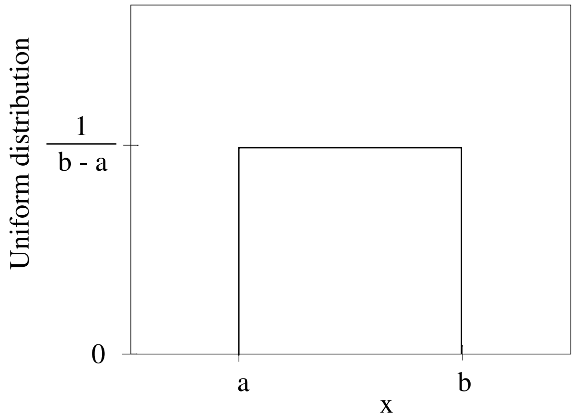

Figure 4.3: Density of the uniform distribution over (a,b)

non-zero entries only on the diagonal. There are as many pairs of these matrices needed as there are jumps in one discrete state and for one fluid place.

Jumps can be jumps up, if the arc indicating the jump is an output arc $a_o \in \mathcal{O}_d \subseteq (\mathcal{T}_E \cup \mathcal{T}_I) \times \mathcal{P}_C$ from a transition to a fluid place, or jumps down, if the arc is an input arc $a_i \in \mathcal{I}_d \subseteq \mathcal{P}_C \times (\mathcal{T}_E \cup \mathcal{T}_I)$ from a fluid place to a transition. Jumps up will be enumerated $\hat{l} = 1, \ldots, \hat{L}$, the jumps down by $\check{l} = 1, \ldots, \check{L}$ the union of both, i.e. all jumps in total are enumerated by $l = 1, \ldots, L$.

Three different probability densities for the jump heights have been implemented. The Dirac impulse is shown in figure 4.2. With $dirac(x)$ the jump height x will happen with probability one if a jump takes place.

The uniform distribution, the density of which is shown in figure 4.3 allows for a broader variety in the jump heights. Uniform(a, b) denotes, that all jump heights between a and b are equally likely, but no height outside that range can happen.

The most flexible is the exponential distribution. Its density is drawn for different values of the parameter λ in figure 4.4. Depending on λ different jump heights are the most likely and more or less variation is possible. Roughly speaking one can say, the larger the expected value, the larger is also the variability of the exponential distribution. So, for large values of λ the exponential distribution comes close to a Dirac impulse with small x, whereas for small values of λ it rather comes close to the uniform distribution with small a and large b.

In summary

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{|S|}), \tag{4.7}$$

$$\mathbf{B} = \mathrm{diag}(b_1, \ldots, b_{|S|}) \tag{4.8}$$

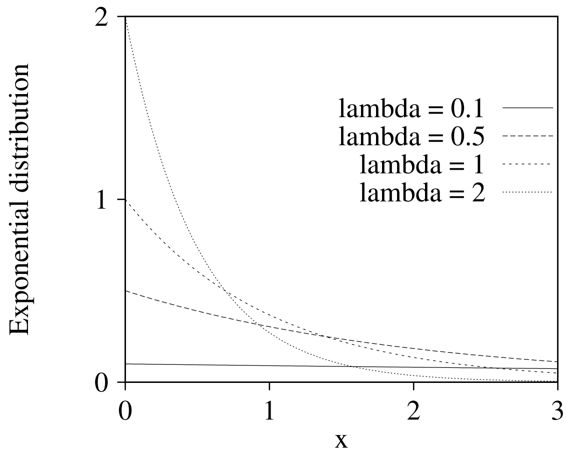

Figure 4.4: Density of the exponential distribution with different values for the parameter λ

where

$$b_s(.) = \begin{cases} dirac(x) \\ uniform(a,b) \\ exponential(\lambda). \end{cases} \tag{4.9}$$

The three distributions have been chosen for their simplicity and diversity, for no stochastic reasons and any other probability densities could be used instead.

Example

The reachability graph of the model in figure 4.1 is shown in figure 4.5. It has only two discrete states and one continuous dimension. The discrete states are defined by the status of the server. The generator matrix can be extracted from the reachability graph and it evaluates to

$$Q = \begin{pmatrix} -\lambda_{t_4} & \lambda_{t_4} \\ \lambda_{t_3} & -\lambda_{t_3} \end{pmatrix}.$$

Since the fluid arcs are not labelled with a flow rate, the change rate function has to be computed from the firing rates of the discrete transitions at the destination and the origin of the fluid arcs.

All transitions in the example in chapter 2 had exponentially distributed firing times, so following (4.4) the parameters for the fluid flow are

$$\mu_1 = \frac{1}{\lambda_{t_1}} - \frac{1}{\lambda_{t_2}}, \quad \sigma_1^2 = \frac{1}{\lambda_{t_1}} + \frac{1}{\lambda_{t_2}} \tag{4.10}$$

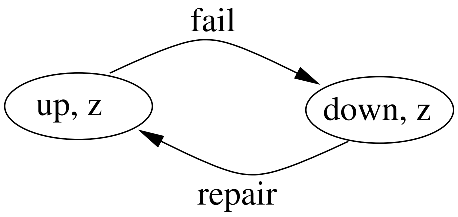

Figure 4.5: Reachability graph of the FSPN queueing model

for the first discrete state, where the server is operational, and

$$\mu_2 = \frac{1}{\lambda_{t_1}}, \quad \sigma_1^2 = \frac{1}{\lambda_{t_1}} \tag{4.11}$$

for the second state, where the server is failed. With the reachability graph and the change rate parameters the underlying Markov process to this simple FSPN is defined. For solving the Markov process boundary conditions still have to be added. The boundary conditions used in this thesis will be presented in a later section in this chapter.

4.3 Examples of Fluid Flow and Jumps in the Continuous Domain

In order to make the vast definitions of the last section more comprehensive, this section contains examples for each of the cases included in the definitions and also examples for model constructions that are not possible.

Figure 4.6 shows an FSPN model where all state changes in the fluid part are just a smooth gliding from one location on the continuous axis to another. The on-off switches represented by the transitions t_3 and t_4 and the places P_2 through P_4 are meant to replace any discrete Petri net that could be attached on either side, i.e. to the transitions t_1 and t_2, but both parts could as well be missing.

The number inside the fluid place denotes the initial value, whereas the number outside is the upper bound. No lower bound is specified, which hence is assumed to be zero. The output arc of transition t_1 is labelled with a rate, that will be used for the computation of the parameters of the fluid change rate. The input arc to t_2 on the other hand is not labelled with a rate, so for this part the firing rate of the exponential transition t_2 will be taken into account.

The reachability graph of the model has four states, it will not be depicted here. We will also omit the generator matrix and focus only on the fluid flow parameters, because the model is very similar to the one in the previous section.

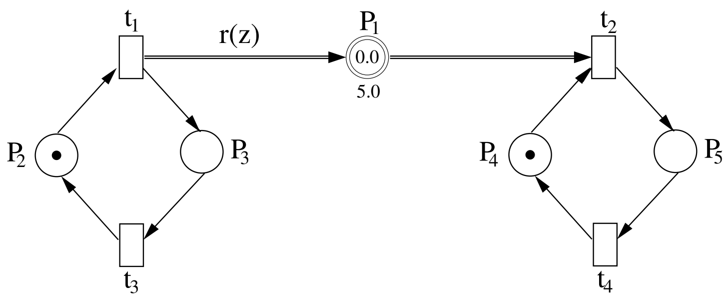

Figure 4.6: Fluid flow along an arc

The expectation and the variance of the change rate function of the model in figure 4.6 are determined as

$$\mu_1(z) = \left(\mu_r(z) - \frac{1}{\lambda_{t_2}(z)}\right) \qquad \text{and} \qquad \sigma_1^2(z) = \left(\sigma_{r_1}^2(z) + \frac{1}{\lambda_{t_2}(z)}\right).$$

$$\mu_2(z) = \left(-\frac{1}{\lambda_{t_2}(z)}\right) \qquad \text{and} \qquad \sigma_2^2(z) = \left(\frac{1}{\lambda_{t_2}(z)}\right).$$

$$\mu_3(z) = (\mu_r(z)) \qquad \text{and} \qquad \sigma_3^2(z) = \left(\sigma_{r_1}^2(z)\right).$$

$$\mu_4(z) = (0) \qquad \text{and} \qquad \sigma_4^2(z) = (0).$$

Hence,

$$\mathbf{M}(z) = \begin{pmatrix} \mu_r(z) - \frac{1}{\lambda_{t_2}(z)} & 0 & 0 & 0 \\ 0 & -\frac{1}{\lambda_{t_2}(z)} & 0 & 0 \\ 0 & 0 & \mu_r(z) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\mathbf{\Sigma}^2(z) = \begin{pmatrix} \sigma_{r_1}^2(z) + \frac{1}{\lambda_{t_2}(z)} & 0 & 0 & 0 \\ 0 & \frac{1}{\lambda_{t_2}(z)} & 0 & 0 \\ 0 & 0 & \sigma_{r_1}^2(z) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The fluid jumps can be applied in a larger variety of situations and will therefore be presented in more than one example. There are three different kinds of situations and for each an example is shown. We distinguish the following cases:

- Jumps can be caused by the firing of an exponential transition at rate $\lambda(\mathbf{z})$ and have a height following the density $b(.)$. An example is shown in figure 4.7. In this case, there is only one discrete state, in which two different jumps can happen, a jump up and a jump down, that happen at the firing rates of t_1 and t_2. The matrices $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ both reduce to a scalar, as well as $\mathbf{B}_1$ and $\mathbf{B}_2$.

$$\mathbf{\Lambda}_1 = \lambda_{t_1}, \qquad \mathbf{\Lambda}_2 = \lambda_{t_2} \quad \text{and} \quad \mathbf{B}_1 = b_1(.), \qquad \mathbf{B}_2 = b_2(.).$$

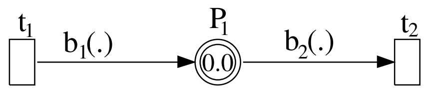

Figure 4.7: First kind of continuous jumps

- The fluid jumps are due to the firing of an immediate discrete transition as is shown in figure 4.8. In this example fluid jumps can happen in two fluid places at the same moment and with the same jump height distribution. So, the firing of the immediate transition can be seen as transporting a certain amount of fluid from one fluid place to the other. Again, one is a jump up, the other a jump down. This becomes important for the analysis in chapter 5. The firing rates of the transitions t_1 and t_3 are chosen to be λ and μ, respectively. The weight of the immediate transition t_2 is equal to 1 since there are no other alternative immediate transitions enabled. The reduced reachability graph is shown in figure 4.9. The states are identified by the number of tokens in P_3 plus the number of tokens in P_4. The immediate transition fires when the model enters states with $\#P_3 + \#P_4$ equal to 1 or 2. The firing of the immediate transition then follows the firing of the exponential transitions t_1 or t_3. Hence, it happens at rate λ or μ as defined in the matrices Λ_{z_1} and Λ_{z_2}, that are identical, since both jumps occur at the firing of the same transition.

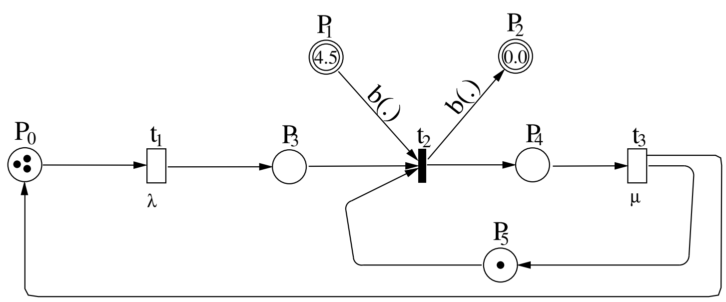

Figure 4.8: Second kind of continuous jumps

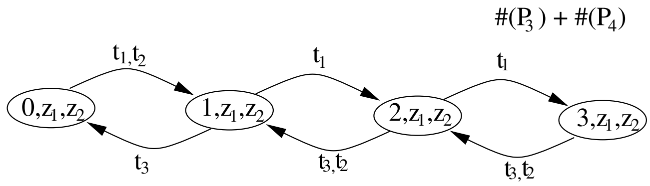

Figure 4.9: Reachability graph of the FSPN containing jumps

$$\mathbf{\Lambda}_{z_1}(\mathbf{z}) = \mathbf{\Lambda}_{z_2}(\mathbf{z}) = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & \mu \end{pmatrix}.$$

The jump heights are chosen to be identical, which is not necessarily the case.

$$\mathbf{B}_1(.) = \mathbf{B}_2(.) = \begin{pmatrix} b(.) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & b(.) & 0 \\ 0 & 0 & 0 & b(.) \end{pmatrix}.$$

- The jumps in the continuous domain are due to the firing of an immediate transition, that does not cause a change of the discrete state as shown in figure 4.10. This situa-

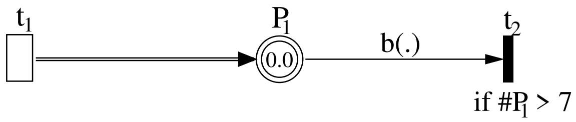

Figure 4.10: Third kind of continuous jumps

tion can not yet be handled numerically because in the underlying stochastic process a jump always occurs in coincidence with a discrete state transition. Therefore jumps of this kind are excluded from the formalism.

4.4 Boundaries

So far the stochastic behaviour of the discrete and the continuous model part have been defined in general. We also have defined upper and lower bounds of the fluid places. At these points specific treatment has to be carried out, because there the behaviour of the model is not the same as in the interior of the continuous domain. For the fluid flow as well as for the fluid jumps special care needs to be taken of the boundaries.

Reflecting boundaries will be used for the fluid flow. This means, that every fluid particle, that reaches a boundary, stays there for a short time and is then rejected back to the interior of the fluid domain. Reflecting barriers are also known for the Random Walk, which can be seen as a discretized diffusion process. More precisely, the diffusion process can be derived as the continuous limit of a Random Walk. The formal derivation of the boundary conditions will be done together with the derivation of the differential equations in chapter 5. For the boundaries of the fluid flow no parameters are needed in the model, since there is no choice or dependency on variables.

In the following other boundary conditions for diffusion processes are listed, that can be found in the literature. Also known from the discrete Random Walk is the *absorbing barrier*, where each particle, that reaches the boundary is absorbed, stays there forever and the process is finished.

Closely related to both the reflecting and the absorbing barrier is the *elementary return* boundary. In an elementary return process a particle is absorbed at the boundary for an exponentially distributed time and is then rejected to an interior point. This point is randomly chosen according to a probability distribution. If the duration of absorption is reduced to zero, the process is called an *instantaneous return* process [Fel54, KT81]. Please note, that the elementary return process has the same characteristics as the jumps that are discussed in this thesis, only that the jumps not only occur at a boundary but at any point in the range of the process.

The elementary return barrier has been proposed as a more correct treatment of an empty queue or fluid place [Gel75]. It has, however, rarely been picked up, most likely because they are more complex and more knowledge of the system is needed.

The *sticky boundary* [KT81] is a reflecting barrier, where the particle stays at the boundary for some positive time before being reflected.

If all flow rates are constant, i.e. not fluid dependent, the *transparent boundaries* [Arn95] can be applied. They simulate an unbounded axis and give an unbiased solution on the regarded domain.

The fluid jumps also have to take care of a limited domain. The jumps in the continuous domain can be seen as a generalization of the elementary return boundary, where the jumps can happen not only at the boundaries but at any place inside the regarded domain. We defined two different methods of handling the boundary: either a jump is forced or, when it is not possible, the former position is preserved. They are abbreviated as *force jump* and *preserve*. For jumps up, with the former a jump height b is sampled and if $z + b > z^{max}$ then $b := z^{max} - z$, where z is the actual fluid level. So, the jump height is reduced to the most possible. In the latter case also a jump height b is sampled from the distribution and if $z + b > z^{max}$ then $b := 0$, so the jump is cancelled. For jumps down the corresponding definition holds at the lower bound. So the jump height function is

$$\text{force\_jump up} \;\cong\; b(.) = \begin{cases} z^{\max} - z & \text{if } z + b > z^{\max} \\ b & \text{else} \end{cases} \tag{4.12}$$

$$\text{preserve up} \;\cong\; b(.) = \begin{cases} 0 & \text{if } z + b > z^{\max} \\ b & \text{else} \end{cases} \tag{4.13}$$

$$\text{force\_jump down} \;\cong\; b(.) = \begin{cases} z - z^{\min} & \text{if } z - b < z^{\min} \\ b & \text{else} \end{cases} \tag{4.14}$$

$$\text{preserve down} \;\cong\; b(.) = \begin{cases} 0 & \text{if } z - b < z^{\min} \\ b & \text{else} \end{cases} \tag{4.15}$$

4.5 Miscellaneous

In this subsection two topics will be discussed that do not gain much attention in this thesis but are too closely related to be neglected completely: the deterministic fluid flow, defining first order FSPNs and simulation of FSPNs.

4.5.1 Deterministic Fluid Flow

When a fluid place represents an actual real valued system component like a clock, the fluid flow might not have any probability distribution, but just a deterministic flow rate [HKNT98]. Then the flow rate is specified by only an expectation and zero variance. It is then possible to solve the first order partial differential equation for the fluid probability distribution function, as has been done in [HKNT98, Hor96]. A rough solution for small first order problems is obtained by forcing probability mass to stay inside the regarded domain [Wol98a]. And it turned out later, that in many cases the second order solution procedure can be used for solving first order models.

However, jumps with deterministic height are possible and indicated by the jump height density function $b(.)$ being concentrated in one point on the continuous axis. This happens when using the Dirac function for $b(.)$ (see the previous section). The occurrence of a jump still is random (with exponentially distributed inter occurrence times).

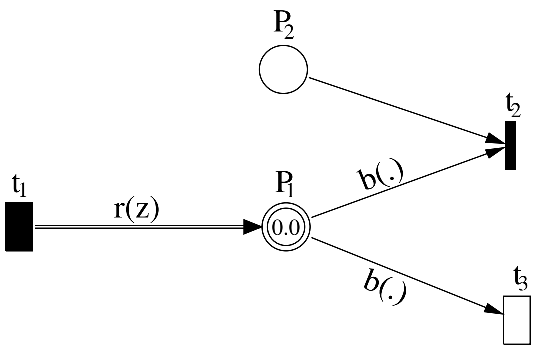

Figure 4.11: Deterministic fluid flow

Definition 4.2 (First order FSPN) A first order FSPN is an FSPN as specified in Definition 4.1 where all flow rates have zero variance. ■

An example of a first order FSPN is shown in figure 4.11, where the black filled rectangle indicates a deterministic transition.

4.5.2 Simulation

From the semantics specified in the previous section it should be possible to start a simulation of an FSPN. But this turns out to be rather difficult. In discrete event simulation a list of possible actions is formulated out of which one is drawn randomly and carried out. This is repeated until some criterion is fulfilled.

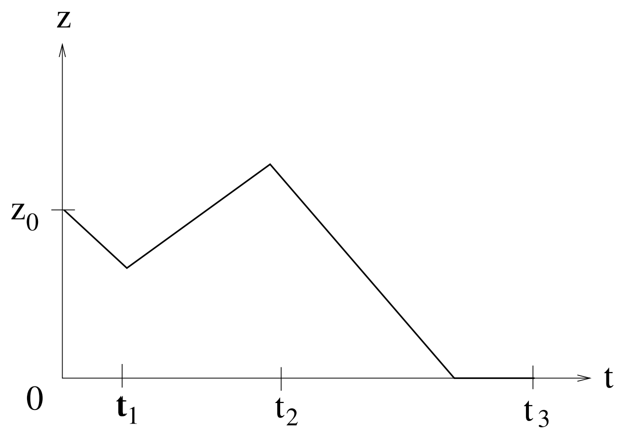

Figure 4.12: Simulated trajectory of the fluid level

To simulate an FSPN the simulation of the discrete part has to be combined with the simulation of the continuous part. So during each discrete action a fluid flow rate has to be specified. Following the principle of simulation the rate has to be drawn from a probability distribution. The question now is, how long this rate will be valid. It would be most natural to preserve the flow rate as long as the discrete action takes (see figure 4.12). This is done in first order models. Keeping the rates constant in a second order model over a longer time period would spoil the random character of the continuous model part [Mac97]. With doing this the model would be changed significantly. The fact that some discrete actions take very long, others a short time makes this effect even worse and harder to control. Hence, one would have to discretize time and regard small time intervals.

Within each period with fixed flow rate another effect has to be taken into account: under the influence of the drawn flow rate the fluid level can reach a boundary (see figure 4.12 and [CNT99]). This instant has to be calculated in advance and the flow rate has to be set to zero there after.

In [CNT99] simulation of a fluid model has been split into a discrete event simulation for the discrete model part and a differential equation for the continuous model part. The differential equation is solved numerically for the time period between two discrete events. If the discrete transition rates may depend on the fluid variables, the discrete model part is represented by a non-homogeneous Markov process, for which simulation techniques have been studied as well [LS79].

Chapter 5

Analysis

In this chapter the differential equations for second order fluid stochastic Petri nets will be derived. They describe the dynamics of the class of models that has been defined in the previous chapter. Second order fluid stochastic Petri nets are Markov models with mixed discrete and continuous state space. In consequence Kolmogorov forward equations can be formulated to derive the partial differential equations. This will be done first for the transient analysis and then for the stationary analysis. For a special case there even exists a closed form steady state solution [Wol97], as will be seen in the last subsection in this chapter.

In the analysis first the reachability graph has to be constructed from a Petri net model. This procedure has been described in chapter 4. At the same time the parameters for the analysis are computed. They are associated with the states of the Petri net rather than the places, transitions and arcs.

All the parameters that are needed for the differential equations have been mentioned earlier. They will be collected and presented together in the first section of this chapter. The transient analysis will be carried out in the second section and the stationary analysis in the third section.

5.1 The Parameters

In this section all the parameters that are needed to define the dynamics of the discrete as well as the continuous part of a fluid stochastic Petri net will be brought together. For the discrete model part these are the state probabilities and the transition rates. For the continuous model part there are as well fluid flow as jump parameters.

The probability of the model being in the discrete state s is denoted π_s, for all discrete states the vector $\boldsymbol{\pi}$ is obtained. The transition rates from one discrete state to an other are the entries of the generator matrix $\mathbf{Q}$ (see eqns. 2.1 and 2.2 on page 14).

If fluid places and hence continuous dimensions are added the probability of being in the discrete state s combined with the density for the continuous dimensions is $\pi_s(z_1, \ldots, z_m)$, if there are m continuous dimensions (and hence m fluid places). The probability vector then is $\boldsymbol{\pi}(\mathbf{z})$. The generator matrix $\mathbf{Q}(\mathbf{z})$ may depend on the continuous values (see eqn. 4.1 on page 34).

The rate at which the fluid level in each fluid place and each discrete state changes is the fluid change rate.

$$R(k, s) = (\mu_{k,s}, \sigma^2_{k,s})$$

(see 4.2 and 4.3 on page 36). The diagonal matrices formed of the flow parameters are $\mathbf{M}(z)$ and $\mathbf{\Sigma}^2(z)$ (see eqns. 4.5 and 4.6 on page 36).

The parameters required for the jumps are the jump rate λ and the jump height $b(.)$. The former is the rate of an exponential distribution, whereas the latter is either the Dirac impulse, the uniform or the exponential distribution. Collected in diagonal matrices for all discrete states the jump rates are $\mathbf{\Lambda}$ and the jump heights are $\mathbf{B}$ (see eqn. 4.7, 4.8 and 4.9 on page 38).

5.2 Transient Analysis

The stochastic process under consideration is a Markov process in continuous time with mixed discrete and continuous state space (see [CM65]).

$$\{(S(t), \mathbf{Z}(t)), t \in \mathbb{R}_0^+\} \tag{5.1}$$

where $S(t)$ is the discrete marking at time t, and $\mathbf{Z}(t)$ is a vector of length m, if there are m fluid places. The components of $\mathbf{Z}(t)$ represent the fluid level in each fluid place. Hence, for each fluid place in the model there is a continuous dimension in the differential equation.

We denote by $\mathbf{\Lambda}(\mathbf{z}) = \sum_{l=1}^{L} \mathbf{\Lambda}_l(\mathbf{z})$ the sum over all jump rates.

At time t the transient probability of being in discrete state i with fluid levels in an infinitesimal environment around z_k, for all fluid places $k \in \mathcal{P}_c$ is called the *fluid density* and is denoted $\pi_s(t, \mathbf{z}) = \partial/\partial\mathbf{z}\ P(S(t) = s, \mathbf{Z}(t) \leq \mathbf{z})$.

The differential equations will first be derived for only one continuous dimension (that is one fluid place in the model) and then generalized to two and more continuous dimensions.

The Chapman-Kolmogorov equation of the process is constituted by looking ahead an infinitesimal step in time

$$
\begin{aligned}
\boldsymbol{\pi}(t+\Delta, z) = \underbrace{\int \boldsymbol{\pi}(t+\Delta, z|t, z')\, \mathbf{diag}\left(\boldsymbol{\pi}(t, z')\right) dz'}_{\text{fluid flow}} + \boldsymbol{\pi}(t, z)\Big(\underbrace{\mathbf{Q}(z)}_{\text{discrete part}} - \underbrace{\boldsymbol{\Lambda}(z)}_{\text{jumps}} \Big)\Delta + \\
\underbrace{\sum_{\hat{l}=1}^{\hat{L}} \int_{z^{\min}}^{z} \boldsymbol{\pi}(t, y)\mathbf{B}_{\hat{l}}(z-y)\, dy\, \boldsymbol{\Lambda}_{\hat{l}}(z)\, \Delta}_{\text{jumps up}} + \\
\sum_{\check{l}=1}^{\check{L}} \underbrace{\int_{z}^{z^{\max}} \boldsymbol{\pi}(t, y)\mathbf{B}_{\check{l}}(y-z)\, dy\, \boldsymbol{\Lambda}_{\check{l}}(z)}_{\text{jumps down}} \Delta + o(\Delta) \quad (5.2)
\end{aligned}
$$

This equation is found by looking at what can happen in a system in a small fraction of time, as it is done for deriving the differential equations for a continuous time Markov chain (CTMC). In the case of the CTMC balance equations are formulated by summing for each state over all the ingoing rates and probabilities minus all the outgoing rates. The same is done here for the discrete state transitions. Another possibility for leaving the mixed discrete and continuous state is to move on to a different continuous state. The first term on the right hand side of (5.2) collects the probability of observing the continuous state z at time $t+\Delta$, conditional on all possible previous continuous states z' which the system could have been in at time t.

In this thesis one more possibility of reaching the continuous state z at time $t+\Delta$ is considered, that are the continuous jumps. Both the last terms in (5.2) collect the probability of having a jump up from position $z-y$ with height y or down from position y with height $z-y$ and both times it is integrated over all possible y smaller than z for the jumps up and greater z for the jumps down.

Now the first integral of eqn. 5.2 is regarded in more detail. Every component $\pi_s(t, z')$ of the vector $\boldsymbol{\pi}(t, z')$ is expanded in a Taylor series around z

$$
\pi_s(t, z') = \pi_s(t, z) + (z'-z)\frac{\partial \pi_s(t, z)}{\partial z} + \frac{(z'-z)^2}{2}\frac{\partial^2 \pi_s(t, z)}{\partial z^2} + o(z'-z)^3 \qquad (5.3)
$$

The second part of the integrand is substituted by the Taylor series

$$
\begin{aligned}
&\int \pi_s(t+\Delta, z|t, z') \left(\pi_s(t, z) + (z'-z)\frac{\partial \pi_s(t, z)}{\partial z} + \frac{(z'-z)^2}{2}\frac{\partial^2 \pi_s(t, z)}{\partial z^2} + o(z'-z)^3 \right) dz' \\
&\quad = \int \pi_s(t+\Delta, z|t, z')\pi_s(t, z) dz' + \int \pi_s(t+\Delta, z|t, z')(z'-z)\frac{\partial \pi_s(t, z)}{\partial z} dz' + \\
&\qquad \int \pi_s(t+\Delta, z|t, z')\frac{(z'-z)^2}{2}\frac{\partial^2 \pi_s(t, z)}{\partial z^2} dz' + o(z'-z)^3 \quad (5.4)
\end{aligned}
$$

Since the integration is over z' all terms not containing z' can be taken outside the integrals. These are the partial derivatives. The conditional density $\pi_s(t+\Delta, z|t, z')$ represents the change in the continuous variable in a time interval of length Δ. It is the distribution of $Z(t+\Delta) - Z(t)$, which is normally distributed with mean $\mu\Delta$ and variance $\sigma^2\Delta$, as has been shown in Section 3.1 for fluid queueing systems. Hence,

$$\pi_s(t+\Delta, z|t, z') = \frac{1}{\sqrt{2\pi\sigma^2\Delta}} \exp\left(\frac{-\left((z-z') - \mu\Delta\right)^2}{2\sigma^2\Delta}\right) \tag{5.5}$$

which is the normal distribution $\Phi(z - z')$. It should be noted, that 5.5 is a normal distribution also with respect to $-z'$ (with mean $(-z + \mu\Delta)$). The normal distribution is substituted in eqn. 5.4

$$\begin{aligned}
\int \pi_s(t+\Delta, z|t, z')\pi_s(t, z')\, dz' = \pi_s(t, z) & \underbrace{\int_\Omega \frac{1}{\sqrt{2\pi\sigma^2\Delta}} \exp\left(\frac{-\left((z-z') - \mu\Delta\right)^2}{2\sigma^2\Delta}\right) dz'}_{=1} \\
+ \frac{\partial \pi_s(t,z)}{\partial z} & \underbrace{\int (z'-z) \frac{1}{\sqrt{2\pi\sigma^2\Delta}} \exp\left(\frac{-\left((z'-z) - \mu\Delta\right)^2}{2\sigma^2\Delta}\right) dz'}_{-\mu(z)\Delta \text{ (negative first moment)}} \\
+ \frac{1}{2}\frac{\partial^2 \pi_s(t,z)}{\partial z^2} & \underbrace{\int_\Omega (z-z')^2 \frac{1}{\sqrt{2\pi\sigma^2\Delta}} \exp\left(\frac{-\left((z-z') - \mu\Delta\right)^2}{2\sigma^2\Delta}\right) dz'}_{\sigma^2(z)\Delta + \mu^2(z)\Delta^2 \text{ (second moment)}} \\
= \pi_s(t,z) - \frac{\partial \pi_s(t,z)}{\partial z}\mu(z)\Delta + \frac{1}{2}\frac{\partial^2 \pi_s(t,z)}{\partial z^2} & (\sigma^2(z)\Delta + \mu^2(z)\Delta^2)
\end{aligned} \tag{5.6}$$

if π is sufficiently smooth so that the following definitions of expectation and variance over the continuous state space hold (see [Gne62], p. 344)

$$\lim_{\Delta \to 0} \frac{1}{\Delta} \int (z'-z)\pi_s(t+\Delta, z|t, z')dz' = \mu(z) \qquad \text{and} \tag{5.7}$$

$$\lim_{\Delta \to 0} \frac{1}{\Delta} \int (z'-z)^2\pi_s(t+\Delta, z|t, z')dz' = \sigma^2(z). \tag{5.8}$$

In vector notation eqn. 5.6 is

$$\boldsymbol{\pi}(t,z) - \frac{\partial \boldsymbol{\pi}(t,z)}{\partial z}\mathbf{M}(z)\Delta + \frac{1}{2}\frac{\partial^2 \boldsymbol{\pi}(t,z)}{\partial z^2}(\boldsymbol{\Sigma}^2(z)\Delta + \mathbf{M}^2(z)\Delta^2) \qquad \forall t, \forall z \in \mathring{\Omega} = (z^{\min}, z^{\max}). \tag{5.9}$$

The first integral in eqn. 5.2 is replaced by eqn. 5.9, $\boldsymbol{\pi}(t,z)$ is subtracted from both sides and the equation is divided by Δ:

$$\frac{\boldsymbol{\pi}(t+\Delta,z)-\boldsymbol{\pi}(t,z)}{\Delta} = -\frac{\partial}{\partial z}\Big(\boldsymbol{\pi}(t,z)\mathbf{M}(z)\Big) + \frac{1}{2}\frac{\partial^2}{2\partial z^2}\Big(\boldsymbol{\pi}(t,z)(\Delta\mathbf{M}^2(z)+\boldsymbol{\Sigma}^2(z))\Big) + \boldsymbol{\pi}(t,z)\Big(\mathbf{Q}(z)-\boldsymbol{\Lambda}(z)\Big) + \sum_{\hat{l}=1}^{\hat{L}}\int_{z^{\min}}^{z}\boldsymbol{\pi}(t,y)\mathbf{B}_{\hat{l}}(z-y)\,dy\,\boldsymbol{\Lambda}_{\hat{l}}(z) + \sum_{\check{l}=1}^{\check{L}}\int_{z}^{z^{\max}}\boldsymbol{\pi}(t,y)\mathbf{B}_{\check{l}}(y-z)\,dy\,\boldsymbol{\Lambda}_{\check{l}}(z) + o(\Delta) \quad (5.10)$$

As Δ approaches zero the second Kolmogorov forward equation is obtained

$$\frac{\partial\boldsymbol{\pi}(t,z)}{\partial t} + \frac{\partial}{\partial z}\Big(\boldsymbol{\pi}(t,z)\mathbf{M}(z)\Big) - \frac{1}{2}\frac{\partial^2}{\partial z^2}\Big(\boldsymbol{\pi}(t,z)\boldsymbol{\Sigma}^2(z)\Big) - \boldsymbol{\pi}(t,z)\Big(\mathbf{Q}(z)-\boldsymbol{\Lambda}(z)\Big) - \sum_{\hat{l}=1}^{\hat{L}}\int_{0}^{z}\boldsymbol{\pi}(t,z-y)\mathbf{B}_{\hat{l}}(y)\,dy\,\boldsymbol{\Lambda}_{\hat{l}}(z) + \sum_{\check{l}=1}^{\check{L}}\int_{z}^{z^{\max}}\boldsymbol{\pi}(t,y)\mathbf{B}_{\check{l}}(z-y)\,dy\,\boldsymbol{\Lambda}_{\check{l}}(z) = 0 \qquad \forall t, \forall z \in \mathring{\Omega}. \quad (5.11)$$

The entries of $\mathbf{M}$ define the drift and the entries of $\boldsymbol{\Sigma}^2$ the variance of the process' behaviour on the continuous axis. Without the jumps and the coupling term $\boldsymbol{\pi}\mathbf{Q}$ the equation is the Fokker-Planck equation of diffusion known from physics.

In the following the parts for the fluid jumps will be neglected. The equations for the fluid flow will be derived for two fluid places and then generalized to an arbitrary number of continuous dimensions.

For two continuous dimensions an infinitesimal forward step in time may include a movement in both continuous directions. Therefore, to specify the probability of being in a certain point in the two-dimensional continuous state space it is necessary to integrate over both directions, regarding in each direction the probability of being at z_i conditional on all possible previous locations z_i' the process could have been in. So (5.2) becomes:

$$\boldsymbol{\pi}(t+\Delta,z_1,z_2) = \iint \boldsymbol{\pi}(t+\Delta,z_1,z_2|t,z_1',z_2')\mathbf{diag}\left(\boldsymbol{\pi}(t,z_1',z_2')\right)\,dz_1'dz_2' + \boldsymbol{\pi}(t,z_1,z_2)\mathbf{Q}(z_1,z_2)\Delta + o(\Delta). \quad (5.12)$$

Again a component s of the vectors will be regarded. For the integral in (5.12) the bivariate normal distribution and the Taylor series expansion for a function in two dimensions is needed. If the random variables $(Z_1 - Z_1')$ and $(Z_2 - Z_2')$ are stochastically independent the bivariate normal distribution is

$$\pi_s(t+\Delta, z_1, z_2 | t, z_1', z_2') = \frac{1}{2\pi\Delta\sigma_1\sigma_2} \exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1 - z_1' - \mu_1\Delta)^2}{\sigma_1^2} + \frac{(z_2 - z_2' - \mu_2\Delta)^2}{\sigma_2^2}\right)\right) \tag{5.13}$$

and (5.13) is the product of the two marginal normal distributions $\Phi(z_1 - z_1')$ and $\Phi(z_2 - z_2')$. If $(Z_1 - Z_1')$ and $(Z_2 - Z_2')$ are correlated (5.13) takes the form

$$\begin{aligned} \pi_s(t+\Delta, z_1, z_2 | t, z_1', z_2') = & \frac{1}{2\pi\Delta\sigma_1\sigma_2\sqrt{1-\rho^2}} \cdot \\ & \exp\left\{-\frac{1}{2\Delta}\left(\frac{1}{1-\rho^2}\right)\left(\frac{((z_1 - z_1') - \mu_1\Delta)^2}{\sigma_1^2}\right)\right\} \cdot \\ & \exp\left\{-\frac{1}{2\Delta}\left(\frac{1}{1-\rho^2}\right)\left(2\rho\frac{(z_1 - z_1') - \mu_1\Delta}{\sigma_1} \cdot \frac{(z_2 - z_2') - \mu_2\Delta}{\sigma_2}\right)\right\}^{-1} \cdot \\ & \exp\left\{-\frac{1}{2\Delta}\left(\frac{1}{1-\rho^2}\right)\left(\frac{((z_2 - z_2') - \mu_2\Delta)^2}{\sigma_2^2}\right)\right\} \end{aligned} \tag{5.14}$$

where ρ is the coefficient of correlation. For notational convenience and because in this thesis only the case of independent processes will be regarded formula (5.13) will be used in the following and $\rho = 0$. The assumption of independent processes is not always realistic. If one fluid flow rate depends on the value of the other fluid variable they are correlated. In this case the covariances must be computed. This is left for future work and at present all covariances are set to zero.

The two-dimensional Taylor series of $\pi_s(t, z_1', z_2')$ around (z_1, z_2) is (see e.g. [Fic64], p. 390f)

$$\begin{aligned} \pi_s(t, z_1', z_2') \quad = \quad & \pi_s(t, z_1, z_2) + (z_1' - z_1)\frac{\partial \pi_s(t, z_1, z_2)}{\partial z_1} + (z_2' - z_2)\frac{\partial \pi_s(t, z_1, z_2)}{\partial z_2} + \\ & \frac{(z_1' - z_1)^2}{2}\frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_1^2} + \frac{(z_2' - z_2)^2}{2}\frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_2^2} + \\ & 2(z_1' - z_1)(z_2' - z_2)\frac{\partial^2 \pi_s(t, z_1, z_2)}{2\partial z_1 \partial z_2} + o(\max(z_i - z_i')^3), \quad i = 1, 2. \end{aligned} \tag{5.15}$$

Expectation and variance exist for each dimension separately, but the density needs to be integrated over both dimensions. Hence, the following limits must exist

$$\lim_{\Delta \to 0} \frac{1}{\Delta} \iint (z_i' - z_i)\pi_s(t+\Delta, z_1, z_2 | t, z_1', z_2') dz_1' dz_2' = \mu_i(z_1, z_2) \qquad i = 1, 2 \tag{5.16}$$

$$\lim_{\Delta \to 0} \frac{1}{\Delta} \iint (z_i' - z_i)^2\pi_s(t+\Delta, z_1, z_2 | t, z_1', z_2') dz_1' dz_2' = \sigma_i^2(z_1, z_2) \qquad i = 1, 2 \tag{5.17}$$

$$\lim_{\Delta\to 0}\frac{1}{\Delta}\iint (z_i'-z_i)(z_j'-z_j)\pi_s(t+\Delta,z_1,z_2|t,z_1',z_2')dz_i'dz_j' = \sigma_{i,j}^2(z_1,z_2) \qquad i,j=1,2 \quad i\neq j. \tag{5.18}$$

Equation (5.16) defines the expectations, (5.17) the variances and (5.18) the covariances. Since the behaviour in both dimensions is assumed to be independent from each other the covariances equal zero.

Then the integral in (5.12) is substituted with (5.13) and (5.15)

$$\begin{aligned}
&\iint \boldsymbol{\pi}(t+\Delta,z_1,z_2|t,z_1',z_2')\mathbf{diag}\left(\boldsymbol{\pi}(t,z_1',z_2')\right)\ dz_1'dz_2' = \\
&\quad\iint \frac{1}{2\pi\Delta\sigma_1\sigma_2}\exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1-z_1'-\mu_1\Delta)^2}{\sigma_1^2}+\frac{(z_2-z_2'-\mu_2\Delta)^2}{\sigma_2^2}\right)\right)\cdot \\
&\quad\quad\left(\pi_s(t,z_1,z_2)+(z_1'-z_1)\frac{\partial\pi_s(t,z_1,z_2)}{\partial z_1}+(z_2'-z_2)\frac{\partial\pi_s(t,z_1,z_2)}{\partial z_2}+\right. \\
&\quad\quad\quad\frac{(z_1'-z_1)^2}{2}\frac{\partial^2\pi_s(t,z_1,z_2)}{\partial z_1^2}+\frac{(z_2'-z_2)^2}{2}\frac{\partial^2\pi_s(t,z_1,z_2)}{\partial z_2^2}+ \\
&\quad\quad\quad\quad\left.2(z_1'-z_1)(z_2'-z_2)\frac{\partial^2\pi_s(t,z_1,z_2)}{2\partial z_1\partial z_2}\right)dz_1'dz_2'
\end{aligned} \tag{5.19}$$

Now the single terms of the summation are integrated and each integral will be handled separately

$$\begin{aligned}
&\iint \frac{1}{2\pi\Delta\sigma_1\sigma_2}\exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1-z_1'-\mu_1\Delta)^2}{\sigma_1^2}+\frac{(z_2-z_2'-\mu_2\Delta)^2}{\sigma_2^2}\right)\right)\cdot \\
&\qquad\qquad \pi_s(t,z_1,z_2)\ dz_1'dz_2' \quad = \quad \underline{\pi_s(t,z_1,z_2)},
\end{aligned} \tag{5.20}$$

because the integral over a density evaluates to one.

$$\begin{aligned}
&\iint \frac{1}{2\pi\Delta\sigma_1\sigma_2}\exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1-z_1'-\mu_1\Delta)^2}{\sigma_1^2}+\frac{(z_2-z_2'-\mu_2\Delta)^2}{\sigma_2^2}\right)\right)\cdot \\
&\qquad (z_1'-z_1)\frac{\partial\pi_s(t,z_1,z_2)}{\partial z_1}dz_1'dz_2' = \frac{\partial\pi_s(t,z_1,z_2)}{\partial z_1}\cdot \\
&\quad\underbrace{\int (z_1'-z_1)\frac{1}{2\pi\Delta\sigma_1\sigma_2}\exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1-z_1'-\mu_1\Delta)^2}{\sigma_1^2}+\frac{(z_2-z_2'-\mu_2\Delta)^2}{\sigma_2^2}\right)\right)dz_1'dz_2'}_{-\mu_1(z_1,z_2)\Delta} \\
&\qquad = \underline{\frac{\partial\pi_s(t,z_1,z_2)}{\partial z_1}(-\mu_1(z_1,z_2)\Delta)}.
\end{aligned} \tag{5.21}$$

And correspondingly

$$\iint \frac{1}{2\pi\Delta\sigma_1\sigma_2}\exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1-z_1'-\mu_1\Delta)^2}{\sigma_1^2}+\frac{(z_2-z_2'-\mu_2\Delta)^2}{\sigma_2^2}\right)\right) \cdot (z_2'-z_2)\frac{\partial\pi_s(t,z_1,z_2)}{\partial z_2}dz_1'dz_2' = \underline{\frac{\partial\pi_s(t,z_1,z_2)}{\partial z_2}(-\mu_2(z_1,z_2)\Delta)} \tag{5.22}$$

because the integrations can be swapped.

The second order terms are treated equivalently

$$\begin{aligned}\iint \frac{1}{2\pi\Delta\sigma_1\sigma_2}\exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1-z_1'-\mu_1\Delta)^2}{\sigma_1^2}+\frac{(z_2-z_2'-\mu_2\Delta)^2}{\sigma_2^2}\right)\right) \cdot \frac{(z_1'-z_1)^2}{2}\frac{\partial^2\pi_s(t,z_1,z_2)}{\partial z_1^2}dz_1'dz_2' = \frac{\partial^2\pi_s(t,z_1,z_2)}{\partial z_1^2} \\ \underbrace{\int \frac{(z_1'-z_1)^2}{2}\frac{1}{2\pi\Delta\sigma_1\sigma_2}\exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1-z_1'-\mu_1\Delta)^2}{\sigma_1^2}+\frac{(z_2-z_2'-\mu_2\Delta)^2}{\sigma_2^2}\right)\right)dz_1'dz_2'}_{\sigma_1^2(z_1,z_2)\Delta+\mu_1^2(z_1,z_2)\Delta^2} \\ = \frac{\partial^2\pi_s(t,z_1,z_2)}{\partial z_1^2}\left(\sigma_1^2(z_1,z_2)\Delta+\mu_1^2(z_1,z_2)\Delta^2\right).\end{aligned} \tag{5.23}$$

With swapping the integrals

$$\begin{aligned}\iint \frac{1}{2\pi\Delta\sigma_1\sigma_2}\exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1-z_1'-\mu_1\Delta)^2}{\sigma_1^2}+\frac{(z_2-z_2'-\mu_2\Delta)^2}{\sigma_2^2}\right)\right) \\ \cdot \frac{(z_2'-z_2)^2}{2}\frac{\partial^2\pi_s(t,z_1,z_2)}{\partial z_2^2}dz_1'dz_2' \\ = \underline{\frac{\partial^2\pi_s(t,z_1,z_2)}{\partial z_2^2}\left(\sigma_2^2(z_1,z_2)\Delta+\mu_2^2(z_1,z_2)\Delta^2\right)}\end{aligned} \tag{5.24}$$

is obtained analogously.

The last term is

$$\iint \frac{1}{2\pi\Delta\sigma_1\sigma_2} \exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1 - z_1' - \mu_1\Delta)^2}{\sigma_1^2} + \frac{(z_2 - z_2' - \mu_2\Delta)^2}{\sigma_2^2}\right)\right) \cdot$$
$$((z_1' - z_1)(z_2' - z_2) + (z_2' - z_2)(z_1' - z_1)) \frac{\partial^2 \pi_s(t, z_1, z_2)}{2\partial z_1 \partial z_2} dz_1' dz_2'$$
$$= \iint \frac{1}{2\pi\Delta\sigma_1\sigma_2} \exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1 - z_1' - \mu_1\Delta)^2}{\sigma_1^2} + \frac{(z_2 - z_2' - \mu_2\Delta)^2}{\sigma_2^2}\right)\right) \cdot$$
$$\frac{\partial^2 \pi_s(t, z_1, z_2)}{2\partial z_1 \partial z_2}(z_1' - z_1)(z_2' - z_2) dz_1' dz_2' +$$
$$\iint \frac{1}{2\pi\Delta\sigma_1\sigma_2} \exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1 - z_1' - \mu_1\Delta)^2}{\sigma_1^2} + \frac{(z_2 - z_2' - \mu_2\Delta)^2}{\sigma_2^2}\right)\right) \cdot$$
$$\frac{\partial^2 \pi_s(t, z_1, z_2)}{2\partial z_2 \partial z_1}(z_1' - z_1)(z_2' - z_2) dz_1' dz_2' =$$

$$= \frac{\partial^2 \pi_s(t, z_1, z_2)}{2\partial z_1 \partial z_2} \cdot$$
$$\underbrace{\iint (z_1' - z_1)(z_2' - z_2) \frac{1}{2\pi\Delta\sigma_1\sigma_2} \exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1 - z_1' - \mu_1\Delta)^2}{\sigma_1^2} + \frac{(z_2 - z_2' - \mu_2\Delta)^2}{\sigma_2^2}\right)\right) dz_1' dz_2'}_{\sigma_{1,2}^2(z_1,z_2)\Delta + \mu_1^2(z_1,z_2)\Delta^2} +$$
$$\frac{\partial^2 \pi_s(t, z_1, z_2)}{2\partial z_2 \partial z_1} \cdot$$
$$\underbrace{\iint (z_1' - z_1)(z_2' - z_2) \frac{1}{2\pi\Delta\sigma_1\sigma_2} \exp\left(-\frac{1}{2\Delta}\left(\frac{(z_1 - z_1' - \mu_1\Delta)^2}{\sigma_1^2} + \frac{(z_2 - z_2' - \mu_2\Delta)^2}{\sigma_2^2}\right)\right) dz_1' dz_2'}_{\sigma_{2,1}^2(z_1,z_2)\Delta + \mu_1^2(z_1,z_2)\Delta^2}$$
$$= \underline{\frac{\partial^2 \pi_s(t, z_1, z_2)}{2\partial z_1 \partial z_2}\left(\sigma_{1,2}^2(z_1, z_2)\Delta + \mu_1^2(z_1, z_2)\Delta^2\right) + \frac{\partial^2 \pi_s(t, z_1, z_2)}{2\partial z_2 \partial z_1}\left(\sigma_{2,1}^2(z_1, z_2)\Delta + \mu_(^2 z_1, z_2)\Delta^2\right)} \tag{5.25}$$

Eqns. (5.20) through (5.25) are filled in eqn. (5.12), giving

$$\pi_s(t+\Delta, z_1, z_2) = \pi_s(t, z_1, z_2) + \frac{\partial \pi_s(t, z_1, z_2)}{\partial z_1}(-\mu_1(z_1, z_2)\Delta) + \frac{\partial \pi_s(t, z_1, z_2)}{\partial z_2}(-\mu_2(z_1, z_2)\Delta) +$$
$$\frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_1^2}\left(\sigma_1^2(z_1, z_2)\Delta + \mu_1^2(z_1, z_2)\Delta^2\right) + \frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_2^2}\left(\sigma_2^2(z_1, z_2)\Delta + \mu_2^2(z_1, z_2)\Delta^2\right) +$$
$$\frac{\partial^2 \pi_s(t, z_1, z_2)}{2\partial z_1 \partial z_2}\left(\sigma_{1,2}^2(z_1, z_2)\Delta + \mu_1^2(z_1, z_2)\Delta^2\right) + \frac{\partial^2 \pi_s(t, z_1, z_2)}{2\partial z_2 \partial z_1}\left(\sigma_{2,1}^2(z_1, z_2)\Delta + \mu_2^2(z_1, z_2)\Delta^2\right) +$$
$$\Big(\boldsymbol{\pi}(t, z_1, z_2)\mathbf{Q}(z_1, z_2)\Big)_s \Delta + o(\Delta). \tag{5.26}$$

Then $\pi_s(t, z_1, z_2)$ is subtracted from both sides and the equation is divided by Δ

$$\frac{\pi_s(t+\Delta, z_1, z_2) - \pi_s(t, z_1, z_2)}{\Delta} = -\frac{\partial \pi_s(t, z_1, z_2)}{\partial z_1}\mu_1(z_1, z_2) - \frac{\partial \pi_s(t, z_1, z_2)}{\partial z_2}\mu_2(z_1, z_2) +$$
$$\frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_1^2}\left(\sigma_1^2(z_1, z_2) + \mu_1^2(z_1, z_2)\Delta\right) + \frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_2^2}\left(\sigma_2^2(z_1, z_2) + \mu_2^2(z_1, z_2)\Delta\right) +$$
$$\frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_1 \partial z_2}\left(\sigma_{1,2}^2(z_1, z_2) + \mu_1^2(z_1, z_2)\Delta\right) + \frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_2 \partial z_1}\left(\sigma_{2,1}^2(z_1, z_2) + \mu_2^2(z_1, z_2)\Delta\right) +$$
$$\Big(\boldsymbol{\pi}(t, z_1, z_2)\mathbf{Q}(z_1, z_2)\Big)_s + o(\Delta). \quad (5.27)$$

As $\Delta \to 0$ the differential equation is obtained

$$\frac{\partial \pi_s(t, z_1, z_2)}{\partial t} = -\sum_{k=1}^{2}\frac{\partial \pi_s(t, z_1, z_2)}{\partial z_k}\mu_k(z_1, z_2) + \sum_{k=1}^{2}\frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_k^2}\sigma_k^2(z_1, z_2) +$$
$$\sum_{i=1}^{2}\sum_{j=1}^{2}\frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_i \partial z_j}\sigma_{i,j}(z_1, z_2) + \Big(\boldsymbol{\pi}(t, z_1, z_2)\mathbf{Q}(z_1, z_2)\Big)_s \quad \forall z_1, z_2 \in \mathring{\Omega}. \quad (5.28)$$

For the process described with (5.28) the normal distribution (5.14) holds.

If the processes on both continuous axes are independent the covariances ($\sigma_{i,j}$) equal zero and what remains of (5.28) is

$$\frac{\partial \pi_s(t, z_1, z_2)}{\partial t} = -\sum_{k=1}^{2}\frac{\partial \pi_s(t, z_1, z_2)}{\partial z_k}\mu_k(z_k) + \sum_{k=1}^{2}\frac{\partial^2 \pi_s(t, z_1, z_2)}{\partial z_k^2}\sigma_k^2(z_k) +$$
$$\Big(\boldsymbol{\pi}(t, z_1, z_2)\mathbf{Q}(z_1, z_2)\Big)_s \quad \forall z_1, z_2 \in \mathring{\Omega}. \quad (5.29)$$

For the general case of m fluid places the m-dimensional normal distribution, the m-dimensional Taylor series and the limits for the mean, variance and covariance are needed. If the process is assumed to be independent in the different dimensions the m-dimensional normal distribution is

$$\pi_s(t+\Delta, z_1, \ldots, z_m | t, z_1', \ldots, z_m') = \prod_{i=1}^{m}\frac{1}{\sqrt{2\pi\Delta\sigma_i}}\exp\left(-\frac{1}{2\Delta}\left(\frac{((z_i - z_i') - \mu_i(z_i)\Delta)^2}{\sigma_i^2(z_i)}\right)\right)$$
$$i = 1, \ldots, m. \quad (5.30)$$

The m-dimensional Taylor-series of $\pi_s(t, z_1', \ldots, z_m')$ around $(z_1, \ldots, z_m)$ with only the first two derivatives is defined as

$$\pi_s(t, z'_1, \dots, z'_m) = \pi_s(t, z_1, \dots, z_m) + \sum_{k=1}^{m} (z'_k - z_k) \frac{\partial \pi_s(t, z_1, \dots, z_m)}{\partial z_k} + \\ \sum_{k=1}^{m} \sum_{j=1}^{m} (z'_k - z_k)(z'_j - z_j) \frac{\partial^2 \pi_s(t, z_1, \dots, z_m)}{\partial z_k \partial z_j} + o(z'_k - z_k). \quad (5.31)$$

The expectation and variance are defined as

$$\lim_{\Delta \to 0} \int \cdots \int (z'_k - z_k) \pi_s(t + \Delta, z_1, \dots, z_m | t, z'_1, \dots, z'_m) dz'_1 \cdots dz'_m = \mu_k(\mathbf{z}) \\ k = 1, \dots, m \quad (5.32)$$

$$\lim_{\Delta \to 0} \int \cdots \int (z'_k - z_k)^2 \pi_s(t + \Delta, z_1, \dots, z_m | t, z'_1, \dots, z'_m) dz'_1 \cdots dz'_m = \sigma_k^2(\mathbf{z}) \\ k = 1, \dots, m \quad (5.33)$$

All the covariances ($\sigma_{i,j}$) are taken equal to zero. For more than two continuous dimensions it becomes practically impossible to specify all covariances since in most cases, one will have to measure or estimate them. It is theoretically straight forward to derive the equations with covariances included, but for their numerical solution more effort is needed.

For m continuous dimensions integration over all dimensions is needed when looking ahead a short moment in time. The jumps are all atomic actions, so they have to be summed. In each term of the summation one component of the π vector is used, therefore the definition of $\mathbf{y}_k$ and $\mathbf{z}_y$ becomes necessary. Both are needed for the integration, where it is integrated only over one component and the others are kept fix. The equivalent to (5.2) and (5.12) in m dimensions is

$$\boldsymbol{\pi}(t + \Delta, \mathbf{z}) = \int \cdots \int \boldsymbol{\pi}(t + \Delta, \mathbf{z} | t, \mathbf{z}') \, \mathbf{diag}\left(\boldsymbol{\pi}(t, \mathbf{z}')\right) dz'_1 \cdots dz'_m + \\ \boldsymbol{\pi}(t, \mathbf{z}) \Big(\mathbf{Q}(\mathbf{z}) - \sum_{k=1}^{m} \mathbf{\Lambda}_k(\mathbf{z}) \Big) \Delta + \sum_{k=1}^{m} \Bigg(\sum_{\hat{l}=1}^{\hat{L}} \Big(\int_{z_k^{\min}}^{z_k} \boldsymbol{\pi}(t, \mathbf{z}_y) \mathbf{B}_{k\hat{l}}(z_k - y) dy \mathbf{\Lambda}_{k\hat{l}}(\mathbf{z}) + \\ \sum_{\check{l}=1}^{\check{L}} \int_{z_k}^{z_k^{\max}} \boldsymbol{\pi}(t, \mathbf{z}_y) \mathbf{B}_{k\check{l}}(y - z_k) dy \; \mathbf{\Lambda}_{k\check{l}}(\mathbf{z}) \Big) \Delta + o(\Delta) \quad (5.34)$$

where

$$\mathbf{z}_y = (z_{y,i})_{i=1,\dots,m} = \begin{cases} y & \text{if } i = k, \\ z_i & \text{otherwise} \end{cases} \quad (5.35)$$

After performing the same steps as it has been done in one and two dimensions the general second Kolmogorov forward equation is in vector notation

$$\begin{aligned}\frac{\partial \boldsymbol{\pi}(t,\mathbf{z})}{\partial t} &= -\sum_{k=1}^{m} \frac{\partial \boldsymbol{\pi}(t,\mathbf{z})}{\partial z_k}\mathbf{M}_k(z_k) + \sum_{k=1}^{m} \frac{\partial^2 \boldsymbol{\pi}(t,\mathbf{z})}{\partial z_k^2}\boldsymbol{\Sigma}_k^2(z_k) + \boldsymbol{\pi}(t,\mathbf{z})\Big(\mathbf{Q}(\mathbf{z}) - \sum_{k=1}^{m}\boldsymbol{\Lambda}_k(\mathbf{z})\Big) \\ &\sum_{k=1}^{m}\Bigg(\sum_{\hat{l}=1}^{\hat{L}}\int_{z_k^{\min}}^{z_k} \boldsymbol{\pi}(t,\mathbf{z}_y)\mathbf{B}_{k\hat{l}}(z_k-y)dy\boldsymbol{\Lambda}_{k\hat{l}}(\mathbf{z}) + \\ \sum_{\check{l}=1}^{\check{L}}\int_{z_k}^{z_k^{\max}} &\boldsymbol{\pi}(t,\mathbf{z}_y)\mathbf{B}_{k\check{l}}(y-z_k)dy\boldsymbol{\Lambda}_{k\check{l}}(\mathbf{z})\Bigg) \quad \forall z_1,\dots,z_m \in \mathring{\Omega} = (z_1^{\min}, z_1^{\max})\times\ldots\times(z_m^{\min}, z_m^{\max}).\end{aligned} \tag{5.36}$$

If the random variables $Z_k - Z_k'$ are correlated a multidimensional form of (5.14) holds and the differential equation is

$$\begin{aligned}\frac{\partial \boldsymbol{\pi}(t,\mathbf{z})}{\partial t} &= -\sum_{k=1}^{m} \frac{\partial \boldsymbol{\pi}(t,\mathbf{z})}{\partial z_k}\mathbf{M}_k(\mathbf{z}) + \sum_{j=1}^{m}\sum_{k=1}^{m} \frac{\partial^2 \boldsymbol{\pi}(t,\mathbf{z})}{\partial z_k \partial z_j}\boldsymbol{\Sigma}_{k,j}^2(\mathbf{z}) + \boldsymbol{\pi}(t,\mathbf{z})\Big(\mathbf{Q}(\mathbf{z}) - \sum_{k=1}^{m}\boldsymbol{\Lambda}_k(\mathbf{z})\Big) \\ &\sum_{k=1}^{m}\Bigg(\sum_{\hat{l}=1}^{\hat{L}}\int_{z_k^{\min}}^{z_k} \boldsymbol{\pi}(t,\mathbf{z}_y)\mathbf{B}_{k\hat{l}}(z_k-y)dy\boldsymbol{\Lambda}_{k\hat{l}}(\mathbf{z}) + \\ &\sum_{\check{l}=1}^{\check{L}}\int_{z_k}^{z_k^{\max}} \boldsymbol{\pi}(t,\mathbf{z}_y)\mathbf{B}_{k\check{l}}(y-z_k)dy\boldsymbol{\Lambda}_{k\check{l}}(\mathbf{z})\Bigg) \quad \forall z_1,\dots,z_m \in \mathring{\Omega}\end{aligned} \tag{5.37}$$

where the variances $\boldsymbol{\Sigma}_{k,k}^2(\mathbf{z})$ have always been written as $\boldsymbol{\Sigma}_k^2(\mathbf{z})$. Note, that in (5.37) the expectation and variance of the fluid change in one place may depend on the values of all other continuous variables.

Equation (5.36) and (5.37) are parabolic partial differential equations of convection-diffusion type.

5.3 Initial and Boundary Conditions

The initial conditions are defined as the configuration the model is in in the beginning. The treatment at the boundaries for the fluid jumps is included in the definition of the jump height functions (see section 4.4), so there are no separate boundary conditions needed. For the fluid flow boundary conditions at the upper and the lower end of the definition range are required.

Definition 5.1 (Initial conditions) Let $\boldsymbol{\pi}_0$ be the vector of initial discrete state probabilities and let $\mathbf{z}_0$ be the vector of initial fluid levels then the inital conditions are

$$\boldsymbol{\pi}(0,\mathbf{z}) = \boldsymbol{\pi}_0 \cdot \delta(\mathbf{z}-\mathbf{z}_0). \tag{5.38}$$

■

The boundary conditions for the fluid flow are determined by the fact that no particle can pass beyond the barrier. This is represented in the density function through the integral over the definition range that evaluates to one.

Theorem 5.2 (Boundary conditions) A reflecting barrier at the lower and the upper bound ($z^{\min}$ and $z^{\max}$) define the boundary condition for the fluid flow [CM65]. A reflecting barrier is defined by the fact that no particle can pass beyond it.

$$\left[\frac{1}{2}\frac{\partial}{\partial z_k}\boldsymbol{\pi}(t,\mathbf{z})\boldsymbol{\Sigma}_k^2(z_k) - \boldsymbol{\pi}(t,\mathbf{z})\mathbf{M}_k(z_k)\right]_{z_k=\{z_k^{\min},z_k^{\max}\}} = 0 \qquad k=1,\dots,m \tag{5.39}$$

<u>Proof:</u> Every density integrates to one. Therefore:

$$\sum_{s=1}^{|\mathcal{S}|}\int\cdots\int_{\Omega} \pi_s(t,\mathbf{z})\ dz_1\cdots dz_m = 1 \tag{5.40}$$

and

$$\frac{\partial}{\partial t}\sum_{s=1}^{|\mathcal{S}|}\int\cdots\int_{\Omega} \pi_s(t,\mathbf{z})\ dz_1\cdots dz_m = \sum_{s=1}^{|\mathcal{S}|}\int\cdots\int_{\Omega} \frac{\partial}{\partial t}\pi_s(t,\mathbf{z})\ dz_1\cdots dz_m = 0. \tag{5.41}$$

We substitute for $\pi_s(t,\mathbf{z})$ with eqn. 5.11

$$\begin{aligned}\sum_{s=1}^{|\mathcal{S}|}\int\cdots\int_{\Omega}\Bigg(\sum_{k=1}^{m}\Bigg(-\frac{\partial}{\partial z_k}\pi_s(t,\mathbf{z})\mu_s(z_k) + \frac{1}{2}\frac{\partial^2}{\partial z_k^2}\pi_s(t,\mathbf{z})\sigma_s^2(z_k) + \left(\boldsymbol{\pi}(t,\mathbf{z})\mathbf{Q}(\mathbf{z})\right)_s\Bigg) + \\ \pi_s(t,\mathbf{z})\sum_{k=1}^{m}\lambda_{k,s}(\mathbf{z}) - \sum_{k=1}^{m}\Bigg(\sum_{\hat{l}=1}^{\hat{L}}\int_{z_k^{\min}}^{z_k}\pi_s(t,\mathbf{z}_y)b_{k,\hat{l},s}(y)dy\lambda_{k,\hat{l},s}(\mathbf{z}) + \\ \sum_{\check{l}=1}^{\check{L}}\int_{z_k}^{z_k^{\max}}\pi_s(t,\mathbf{z}_y)b_{k,\check{l},s}(y-z_k)dy\Bigg)\lambda_{k,\check{l},s}(\mathbf{z})\Bigg)dz_1\cdots dz_m = 0.\end{aligned} \tag{5.42}$$

Now the single terms of the sum are integrated

$$\sum_{s=1}^{|S|}\int_{\Omega}\cdots\int\sum_{k=1}^{m}\left(-\frac{\partial}{\partial z}\pi_s(t,\mathbf{z})\mu_s(\mathbf{z})+\frac{1}{2}\frac{\partial^2}{\partial z^2}\pi_s(t,\mathbf{z})\sigma_s^2(\mathbf{z})\right)dz_1\cdots dz_m+$$
$$\sum_{s=1}^{|S|}\int_{\Omega}\cdots\int\sum_{k=1}^{m}\left(\boldsymbol{\pi}(t,\mathbf{z})\mathbf{Q}(\mathbf{z})\right)_s dz_1\cdots dz_m+$$
$$\sum_{s=1}^{|S|}\int_{\Omega}\cdots\int\Bigg(\pi_s(t,\mathbf{z})\sum_{k=1}^{m}\lambda_k(\mathbf{z})-\sum_{k=1}^{m}\Bigg(\sum_{\hat{l}=1}^{\hat{L}}\int_{z_k^{\min}}^{z_k}\pi_s(t,\mathbf{z}_y)b_{k,\hat{l},s}(y)dy\lambda_{k,\hat{l},s}(\mathbf{z})+$$
$$\sum_{\check{l}=1}^{\check{L}}\int_{z_k}^{z_k^{\max}}\pi_s(t,\mathbf{z}_y)b_{k,\check{l},s}(y-z_k)dy\lambda_{k,\check{l},s}(\mathbf{z})\Bigg)\Bigg)dz_1\cdots dz_m=0. \quad (5.43)$$

The coupling term $\boldsymbol{\pi Q}$ as well as all the terms connected to the jumps evaluate to zero as is shown in the following.

$$\sum_{s=1}^{|S|}\int_{\Omega}\cdots\int\sum_{k=1}^{m}\left(\boldsymbol{\pi}(t,\mathbf{z})\mathbf{Q}(\mathbf{z})\right)_s dz_1\cdots dz_m= \quad (5.44)$$

$$=\int_{\Omega}\cdots\int\textstyle\sum_{k=1}^{m}\sum_{s=1}^{|S|}\left(\boldsymbol{\pi}(t,\mathbf{z})\mathbf{Q}(\mathbf{z})\right)_s dz_1\cdots dz_m \quad (5.45)$$

$$=\int_{\Omega}\cdots\int\textstyle\sum_{k=1}^{m}\underbrace{\displaystyle\sum_{s=1}^{|S|}\sum_{i=1}^{|S|}\pi_i q_{i,s}(\mathbf{z})}_{=0}dz_1\cdots dz_m=0 \quad (5.46)$$

because the rows of $\mathbf{Q}$ sum to zero. The elements of $\mathbf{Q}$ are denoted $q_{i,s}$.

The jumps only shift probability mass along the continuous axis and so

$$\sum_{s=1}^{|S|}\int_{\Omega}\cdots\int\lambda_{k,l,s}(\mathbf{z})\underbrace{\left(\pi_s(t,\mathbf{z})+\int_{z_k^{\min}}^{z}\pi_s(t,\mathbf{z}_k)b_{k,l,s}(z_k-y)dy\right)}_{=0}dz_1\cdots dz_m=0 \quad (5.47)$$

for all $\hat{l},\check{l}$ and k (jumps up and down for all fluid places). Hence eqn. 5.43 reduces to

$$\sum_{s=1}^{|S|}\int_{\Omega}\cdots\int\sum_{k=1}^{m}\left(-\frac{\partial}{\partial z}\left(\pi_s(t,\mathbf{z})\mu_s(z_k)\right)+\frac{1}{2}\frac{\partial^2}{\partial z^2}\left(\pi_s(t,\mathbf{z})\sigma_s^2(z_k)\right)\right)dz_1\cdots dz_m=0 \quad (5.48)$$

which is equivalent to

$$\sum_{s=1}^{|S|}\sum_{k=1}^{m}\int_{z_1^{\min}}^{z_1^{\max}}\cdots\int_{z_m^{\min}}^{z_m^{\max}}\frac{\partial}{\partial z}\left[-\pi_s(t,\mathbf{z})\mu_s(z_k)+\frac{1}{2}\frac{\partial}{\partial z}\left(\pi_s(t,\mathbf{z})\sigma_s^2(\mathbf{z})\right)\right]dz_1\cdots dz_m=0. \tag{5.49}$$

Eqn. 5.49 evaluates to

$$\sum_{s=1}^{|S|}\sum_{k=1}^{m}\left[-\pi_s(t,\mathbf{z})\mu_s(z_k)+\frac{1}{2}\frac{\partial}{\partial z_k}\pi_s(t,\mathbf{z})\sigma_s^2(z_k)\right]\begin{matrix} z_1^{\max} \\ \vdots \\ z_m^{\max} \\ z_1^{\min} \\ \vdots \\ z_m^{\min}\end{matrix} = 0. \tag{5.50}$$

But the stronger conditions

$$\frac{1}{2}\frac{\partial}{\partial z_k}\pi_s(t,\mathbf{z})\sigma_s^2(z_k^{\min})-\pi_s(t,\mathbf{z})\mu_s(z_k^{\min})\bigg|_{\begin{matrix} z_1^{\min} \\ \vdots \\ z_m^{\min}\end{matrix}} = 0 \tag{5.51}$$

and

$$\frac{1}{2}\frac{\partial}{\partial z_k}\pi_s(t,\mathbf{z})\sigma_s^2(z_k^{\max})-\pi_s(t,\mathbf{z})\mu_s(z_k^{\max})\bigg|_{\begin{matrix} z_1^{\max} \\ \vdots \\ z_m^{\max}\end{matrix}} = 0 \tag{5.52}$$

are imposed. ∎

5.4 Stationary Analysis

The equations derived for the transient analysis are assumed to have the property of converging towards a stationary value if the final point in time they are solved for is sufficiently large. So, to derive the stationary solution this limit is carried out theoretically and (5.36) is regarded as time approaches infinity.

For the vector of probability densities we look at

$$\lim_{t\to\infty}\boldsymbol{\pi}(t,\mathbf{z})=\boldsymbol{\pi}(\mathbf{z}). \tag{5.53}$$

The same limit is carried out for the differential equation

$$\lim_{t\to\infty} \quad \frac{\partial \boldsymbol{\pi}(t,\mathbf{z})}{\partial t} + \sum_{k=1}^{m} \frac{\partial}{\partial z_k}\Big(\boldsymbol{\pi}(t,\mathbf{z})\mathbf{M}_k(z_k)\Big) - \sum_{k=1}^{m} \frac{1}{2}\frac{\partial^2}{\partial z_k^2}\Big(\boldsymbol{\pi}(t,\mathbf{z})\boldsymbol{\Sigma}_k^2(z_k)\Big) -$$
$$\boldsymbol{\pi}(t,\mathbf{z})\Big(\mathbf{Q}(\mathbf{z}) - \sum_{k=1}^{m}\boldsymbol{\Lambda}_k(\mathbf{z})\Big) - \sum_{k=1}^{m}\Bigg(\sum_{\hat{l}=1}^{\hat{L}}\int_{z_k^{\min}}^{z_k} \boldsymbol{\pi}(t,\mathbf{z}_k)\mathbf{B}_{k\hat{l}}(z_k-y)dy\boldsymbol{\Lambda}_{k\hat{l}}(\mathbf{z}) +$$
$$\sum_{\check{l}=1}^{\check{L}}\int_{z_k}^{z_k^{max}} \boldsymbol{\pi}(t,\mathbf{z}_y)\mathbf{B}_{k\check{l}}(y-z_k)dy\;\boldsymbol{\Lambda}_{k\check{l}}(\mathbf{z})\Bigg) = 0,$$

giving as the stationary equation

$$\sum_{k=1}^{m} \frac{\partial}{\partial z_k}\Big(\boldsymbol{\pi}(\mathbf{z})\mathbf{M}_k(z_k)\Big) - \sum_{k=1}^{m} \frac{1}{2}\frac{\partial^2}{\partial z_k^2}\Big(\boldsymbol{\pi}(\mathbf{z})\boldsymbol{\Sigma}_k^2(z_k)\Big) - \boldsymbol{\pi}(\mathbf{z})\Big(\mathbf{Q}(\mathbf{z}) - \sum_{k=1}^{m}\boldsymbol{\Lambda}_k(\mathbf{z})\Big) -$$
$$\sum_{k=1}^{m}\Bigg(\sum_{\hat{l}=1}^{\hat{L}}\int_{z_k^{\min}}^{z_k} \boldsymbol{\pi}(\mathbf{z}_k)\mathbf{B}_{k\hat{l}}(z_k-y)dy\boldsymbol{\Lambda}_{k\hat{l}}(\mathbf{z}) +$$
$$\sum_{\check{l}=1}^{\check{L}}\int_{z_k}^{z_k^{max}} \boldsymbol{\pi}(\mathbf{z}_y)\mathbf{B}_{k\check{l}}(y-z_k)dy\;\boldsymbol{\Lambda}_{k\check{l}}(\mathbf{z})\Bigg) = 0. \quad (5.54)$$

Equation (5.54) is a partial differential equation only in the space dimensions, so it is an equation in one dimension less than the transient differential equation is. It should therefore be easier to solve.

The boundary conditions are obtained similarly. The reflecting barrier boundary conditions in (5.51) and (5.52) are also valid when the process has reached equilibrium. They can be evaluated for $\boldsymbol{\pi}(\mathbf{z})$ or their limit $t \to \infty$ is taken as it is done for the differential equation. The result always is

$$\frac{1}{2}\frac{\partial}{\partial z_k}\pi_s(\mathbf{z})\sigma_s^2(z_k^{\min}) - \pi_s(\mathbf{z})\mu_s(z_k^{\min})\Big|_{\substack{z_1^{\min}\\ \vdots \\ z_m^{\min}}} = 0 \quad (5.55)$$

and

$$\frac{1}{2}\frac{\partial}{\partial z_k}\pi_s(\mathbf{z})\sigma_s^2(z_k^{\max}) - \pi_s(\mathbf{z})\mu_s(z_k^{\max})\Big|_{\substack{z_1^{\max}\\ \vdots \\ z_m^{\max}}} = 0. \quad (5.56)$$

5.5 Miscellaneous

In this section related topics and interesting special cases will be discussed. At first an idea from [HKNT98] is picked up, where for first order FSPNs a transformation is carried out, that maps the infinite real positive axis on the interval $[0, 1]$. The same transformation can be applied to second order models.

Then there will be a paragraph on first order models, because they are very closely related and finally the analytical stationary solution for the special case of only one fluid place, no jumps and constant flow parameters is derived.

5.5.1 The Differential Equations in a Transformed Domain

A difficulty in the differential equations derived in the previous section is, that they are supported on an infinite domain, which for the numerical solution of course has to be truncated. So, already in the Petri net model one has to decide what range a fluid place shall have.

There is a way of mapping the infinite axis $[0, \infty]$ on the interval $[0, 1]$ by transforming variables. This has already been done in [HKNT98]. The variable z is replaced by $x = 1 - \mathrm{e}^{-z}$.

When substituting z by x, where $z = \varphi(x)$ the first derivative is replaced as follows (see [BS68])

$$\frac{d\,\pi(t,z)}{d\,z} = \frac{1}{\varphi'(x)}\frac{d\,\pi(t,x)}{d\,x} \tag{5.57}$$

and the second derivative

$$\frac{d^2\,\pi(t,z)}{d\,z^2} = \frac{1}{(\varphi'(x))^3}\left(\varphi'(x)\frac{d^2\,\pi(t,x)}{d\,x^2} - \varphi''(x)\frac{d\,\pi(t,x)}{d\,x}\right). \tag{5.58}$$

And $\varphi(x) = -\ln(1-x)$ so after having carried out the substitutions the differential equation (5.11 on page 51) takes the form:

$$\begin{aligned}
&\frac{\partial\boldsymbol{\pi}(t,\mathbf{x})}{\partial t} + \sum_{k=1}^{m}\frac{\partial}{\partial x_k}\Big(\boldsymbol{\pi}(t,\mathbf{x})\mathbf{M}_k(x_k)\Big)\cdot 2(1-x_k) - \sum_{k=1}^{m}\frac{1}{2}\frac{\partial^2}{\partial x_k^2}\Big(\boldsymbol{\pi}(t,\mathbf{x})\boldsymbol{\Sigma}_k^2(x_k)\Big)\cdot(1-x_k)^2 - \\
&\quad\boldsymbol{\pi}(t,\mathbf{x})\Big(\mathbf{Q}(\mathbf{x}) - \sum_{k=1}^{m}\boldsymbol{\Lambda}_k(\mathbf{x})\Big) - \sum_{k=1}^{m}\Bigg(\sum_{\hat{l}=1}^{\hat{L}}\int_{x_k^{\min}}^{x_k}\boldsymbol{\pi}(t,\mathbf{x}_y)\mathbf{B}_{k\hat{l}}(x_k-y)dy\boldsymbol{\Lambda}_{k\hat{l}}(\mathbf{x}) + \\
&\qquad\qquad\sum_{\check{l}=1}^{\check{L}}\int_{z_k}^{x_k^{\max}}\boldsymbol{\pi}(t,\mathbf{x}_y)\mathbf{B}_{k\check{l}}(y-x_k)dy\boldsymbol{\Lambda}_{k\check{l}}(\mathbf{x})\Bigg) = 0 \qquad x\in(0,1).
\end{aligned} \tag{5.59}$$

where $\mathbf{x}_y$ is defined analogously to $\mathbf{z}_y$ in (5.35).

This transformation removes the difficulty of placing an appropriate boundary, when the solution over the whole axis is desired. Another way of obtaining the same result is to impose transparent boundaries. But transparent boundaries can only be used if the coefficients in (5.11), (5.29) and (5.36) are constant. This means, that mean and variance of the change in a fluid place may not be dependent on the fluid levels. Then the transparent boundary condition represents the behaviour of the system on an infinite axis.

Note, that (5.59) is valid only for second order models. For first order models the transformation needs to be carried out separately, it is not obtained by simply dropping the second order term.

The boundary conditions have to be transformed in an analogous way. If the upper bound is at $x_k = 1$ the derivative term vanishes and at the boundary the density $\pi_s(t, \mathbf{x})$ becomes zero, as expected. If the lower bound is at $x_k = 0$ the transformation factor disappears. So, at the boundaries the following equations hold

$$\left.\frac{1-x_k}{2}\frac{\partial}{\partial x_k}\pi_s(t,\mathbf{x}) - \pi_s(t,\mathbf{x})\right|_{\substack{x_1^{\min}\\ \vdots \\ x_m^{\min}}} = 0 \tag{5.60}$$

and

$$\left.\frac{1-x_k}{2}\frac{\partial}{\partial x_k}\pi_s(t,\mathbf{x}) - \pi_s(t,\mathbf{x})\right|_{\substack{x_1^{\max}\\ \vdots \\ x_m^{\max}}} = 0. \tag{5.61}$$

Since in (5.59) and the boundary conditions (5.60) and (5.61) the same amount of information is contained in a much smaller continuous domain the solution by discretization will need a much smaller stepsize, so that in the end, to get a certain quality in the results, the amount of work needed will be the same. But still, one needs not to worry about where to place the boundaries.

5.5.2 The Differential Equations for First Order Models

First order models are characterized by fluid flow, that has no variation and is in consequence specified only by the expectation. The variance always equals zero. States that correspond to first order models arise naturally in second order models if in some states the fluid flow is deterministic and there is no variance. So, indeed both kinds of models must go together, since they can be represented in different states within the same model. First order models have been studied in more detail in [HKNT98, Hor98].

It has turned out that the numerical algorithm for the solution of the second order equations as it is derived in the next chapter also works well if the variances are zero. This will be demonstrated in an application in section 7.3.

If in 5.36 the variances are set to zero the differential equation for the probability densities is obtained

$$\frac{\partial \boldsymbol{\pi}(t,\mathbf{z})}{\partial t} + \sum_{k=1}^{m} \frac{\partial}{\partial z_k}\Big(\boldsymbol{\pi}(t,\mathbf{z})\mathbf{M}_k(\mathbf{z})\Big) - \boldsymbol{\pi}(t,\mathbf{z})\Big(\mathbf{Q}(\mathbf{z}) - \sum_{k=1}^{m}\boldsymbol{\Lambda}_k(\mathbf{z})\Big) -$$
$$\sum_{k=1}^{m}\Bigg(\sum_{\hat{l}=1}^{\hat{L}}\int_{z_k^{\min}}^{z_k} \boldsymbol{\pi}(t,\mathbf{z}_k)\mathbf{B}_{k\hat{l}}(z_k-y)dy\boldsymbol{\Lambda}_{k\hat{l}}(\mathbf{z}) +$$
$$\sum_{\check{l}=1}^{\check{L}}\int_{z_k}^{z_k^{\max}} \boldsymbol{\pi}(t,\mathbf{z}_y)\mathbf{B}_{k\check{l}}(y-z_k)dy\boldsymbol{\Lambda}_{k\check{l}}(\mathbf{z})\Bigg) = 0. \quad (5.62)$$

The initial conditions in first order models are defined by the initial configuration of the model, so they are the same as in second order models.

$$\boldsymbol{\pi}(0,\mathbf{z}) = \boldsymbol{\pi}_0 \cdot \delta(\mathbf{z}-\mathbf{z}_0) \quad (5.63)$$

where $\boldsymbol{\pi}_0$ is the vector of initial discrete state probabilities, $\mathbf{z}_0$ the vector of initial fluid levels and δ the Dirac impulse.

For first order differential equations only one boundary condition is needed. It can be derived analogously to the boundary conditions for second order models as it has been done in Section 5.3.

If the convection parameter μ is constant, then a boundary condition on only one end of the definition range is needed. It is

$$\pi_s(t,\mathbf{z})\Big|_{z_k=z_k^{\min}} = 0 \qquad \text{if } \mu_{k,s} \geq 0 \quad (5.64)$$

and otherwise

$$\pi_s(t,\mathbf{z})\Big|_{z_k=z_k^{\max}} = 0 \qquad \text{if } \mu_{k,s} < 0. \quad (5.65)$$

If μ is fluid-dependent on the other hand, one has to check which one of the above cases holds near the boundary. In [Hor98] instead of the probability density the probability distribution function has been solved for. There the question of boundary conditions is always easier to answer, because for every distribution function F it is well known that

$$F(z^{\min}) = 0 \qquad \text{and} \qquad F(z^{\max}) = 1.$$

But the differential equation for the distribution function includes integrals and is therefore more laborious to solve.

Simulation of first order models has been done recently [CNT99]. It is easier than the simulation of second order models as it has been discussed in Section 4.5.2.

5.5.3 The Analytical Steady State Solution for a Special Case

As it could be seen in Chapter 3 there are diffusion models for which analytical solutions do exist. For the most general diffusion model it is not possible to find a closed form solution. But for the special case of constant coefficients, that are state independent expectation and variance of the fluid flow, in models with only one fluid place and no jumps the analytical steady state solution will be derived in the following.

Each component of the vector is integrated separately and then the stationary state probability in the second order FSPN is

$$\boldsymbol{\pi} = \int \boldsymbol{\pi}(z)\, dz.$$

For the special case considered here (5.36 on page 58) reduces to:

$$\frac{\partial \boldsymbol{\pi}(z)}{\partial z}\mathbf{M} - \frac{1}{2}\frac{\partial^2 \boldsymbol{\pi}(z)}{\partial z^2}\boldsymbol{\Sigma}^{\mathbf{2}} - \boldsymbol{\pi}\mathbf{Q} = \mathbf{0}. \tag{5.66}$$

In state of equilibrium $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ holds. It is therefore possible to first compute the vector of state probabilities $\boldsymbol{\pi}$ by solving $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$. The fluid densities then are determined by the solutions of

$$\frac{\partial \boldsymbol{\pi}(z)}{\partial z}\mathbf{M} - \frac{1}{2}\frac{\partial^2 \boldsymbol{\pi}(z)}{\partial z^2}\boldsymbol{\Sigma}^{\mathbf{2}} = \mathbf{0}. \tag{5.67}$$

In the following it is assumed that $\mu_s \neq 0$. Integration of (5.67) gives for each component π_s of the vector $\boldsymbol{\pi}$

$$\int \left(\frac{\partial \pi_s(z)}{\partial z}\,\mu_s - \frac{1}{2}\frac{\partial^2 \pi_s(z)}{\partial z^2}\,\sigma_s^2 \right) dz = 0 \tag{5.68}$$

$$\Leftrightarrow \pi_s(z)\,\mu_s - \frac{1}{2}\frac{\partial \pi_s(z)}{\partial z}\,\sigma_s^2 \Bigg|_{z^{min}}^{z^{\max}} = C'. \tag{5.69}$$

The left hand side of (5.69) is the boundary condition (5.51) and (5.52) and it therefore follows that $C' = 0$. The solution of

$$\pi_s(z)\,\mu_s - \frac{1}{2}\frac{\partial \pi_s(z)}{\partial z}\,\sigma_s^2 \Bigg|_{z^{min}}^{z^{\max}} = 0 \tag{5.70}$$

is known to be

$$\pi_s(z) = C \cdot \exp\left(\frac{2\mu_s}{\sigma_s^2} z \right) \tag{5.71}$$

where C is a constant of integration. To determine the value of C both sides of (5.71) are integrated

$$\pi_s = \int C \cdot \exp\left(\frac{2\mu_s}{\sigma_s^2} z\right) \, dz. \tag{5.72}$$

Hence

$$C = -\pi_s \cdot \frac{2\mu_s}{\sigma_s^2} \left(\exp\left(\frac{2\mu_s}{\sigma_s^2} z^{\min}\right) - \exp\left(\frac{2\mu_s}{\sigma_s^2} z^{\max}\right) \right) \tag{5.73}$$

and it follows for $\pi_s(z)$

$$\pi_s(z) = -\pi_s \cdot \frac{2\mu_s}{\sigma_s^2} \left(\exp\left(\frac{2\mu_s}{\sigma_s^2} (z^{\min} + z)\right) - \exp\left(\frac{2\mu_s}{\sigma_s^2} (z^{\max} + z)\right) \right) \tag{5.74}$$

The following cases can be distinguished

- $\mu_s < 0$: Then (5.74) holds on the bounded interval. And

$$\pi_s(z) = -\pi_s \cdot \frac{2\mu_s}{\sigma_s^2} \exp\left(\frac{2\mu_s}{\sigma_s^2} z\right) \tag{5.75}$$

  for $z^{\min} = 0$ and $z^{\max} = \infty$.

- $\mu_s > 0$: A steady state solution only exists on the bounded interval $[z^{\min}, z^{\max}]$ and it is given by (5.74).

- $\mu_s = 0$: In this case (5.67) reduces to

$$\frac{\sigma_s^2}{2} \frac{\partial^2 \pi_s(z)}{\partial z^2} = 0 \tag{5.76}$$

  and (5.71) becomes

$$\pi_s(z) = C \cdot \frac{2}{\sigma_s^2}. \tag{5.77}$$

  The constant C again is determined by

$$\begin{aligned} \int_{z^{\min}}^{z^{\max}} C \cdot \frac{2}{\sigma_s^2} \, dz &= \pi_s \\ \Rightarrow \qquad \pi_s(z) &= \pi_s \cdot \frac{1}{z^{\max} - z^{\min}}. \end{aligned} \tag{5.78}$$

  Equation (5.78) is only defined for a finite value of $z^{\max}$. Then it is a constant value over the whole z-axis. Since there is no drift, the variance makes that in steady state all probability mass is distributed equally over the definition range.

It could be seen that a steady state distribution on an infinite interval only exists if $\mu_s < 0$, so in both other cases only finite intervals can be considered.

This very simple model will not be used very frequently, but still when ever it is possible one should omit numerical procedures and use analytical solutions.

Chapter 6

Numerical Methods

In this chapter the numerical methods, used to solve the equations derived in the previous chapter, will be discussed. In chapter 7 these methods will be applied for the analysis of exemplary models. The chapter is divided into several parts. On the top level there are two sections, one about methods for the transient analysis and the other about the stationary analysis. The stationary analysis has been neglected in this thesis, only one method will be presented, that is easy to implement but optimal neither with respect to run time nor to memory usage.

Most of this chapter is occupied by the transient analysis. A very powerful and flexible method, the theta-scheme, is described. It includes the fully implicit, the fully explicit and the Crank-Nicolson scheme [MM94] as special cases.

This thesis aims at providing a tool that is useful for many standard problems. It is far beyond the scope of this work to provide the number of algorithms, that is needed in order to be able to handle special situations in an optimal way. The theta-method accomplishes this task, but it is not suitable for extreme cases such as very large drift and variance, or convection dominated problems, that require specially tailored discretization schemes [KHR89].

6.1 Discretization Methods for the Transient Analysis

The transient analysis of models without jumps in the continuous part has been carried out for models with one and with two fluid places. The discretization must be developed separately for the one and the two-dimensional problem and both are described in an own subsection.

The jumps in the continuous domain have yet been applied only to models with one fluid place. Their numerical treatment consists of solving the convolution integrals in 5.11, which will be done in a separate subsection.

Hence, the first section is divided in three subsections, the first two for one-dimensional models without and with jumps, the third for one-dimensional problems. For models without jumps the transient analysis has been developed in [Zis98].

6.1.1 The Discretization Scheme for one Continuous Dimension

In this subsection only models with one fluid place will be considered, the fluid jumps and the second continuous dimension will follow in later subsections. So (5.11) reduces to

$$\frac{\partial \boldsymbol{\pi}(t,z)}{\partial t} + \frac{\partial}{\partial z}\Big(\boldsymbol{\pi}(t,z)\mathbf{M}(z)\Big) = \frac{1}{2}\frac{\partial^2}{\partial z^2}\Big(\boldsymbol{\pi}(t,z)\boldsymbol{\Sigma}^2(z)\Big) + \boldsymbol{\pi}(t,z)\mathbf{Q}(z)$$
$$\forall t, \forall z \in \overset{\circ}{\Omega} = (z^{\min}, z^{\max}). \quad (6.1)$$

subject to the boundary condition

$$\left[\frac{1}{2}\frac{\partial}{\partial z}\boldsymbol{\pi}(t,z)\Sigma^2(z) - \boldsymbol{\pi}(t,z)\mathbf{M}(z)\right]_{z=\{z^{\min},z^{\max}\}} \quad (6.2)$$

and the initial condition

$$\boldsymbol{\pi}(0,z) = \boldsymbol{\pi}_0\delta(z-z_0). \quad (6.3)$$

For the differentiation terms the theta-method is used, which has as special cases the fully implicit ($\theta = 1$), explicit ($\theta = 0$) and the Crank-Nicolson ($\theta = \frac{1}{2}$) scheme. The first space derivative is discretized in an upwind scheme in order to improve the stability of the method. The coupling term $\boldsymbol{\pi}\mathbf{Q}$ will be discretized explicitly so the equation for each discrete state can be solved separately before going on to the next time step.

For one state the equation that must be discretized is

$$\frac{\partial \pi_s(t,z)}{\partial t} + \frac{\partial}{\partial z}\Big(\pi(t,z)\mu_s(z)\Big) = \frac{1}{2}\frac{\partial^2}{\partial z^2}\Big(\pi(t,z)\sigma_s^2(z)\Big) + \Big(\boldsymbol{\pi}(t,z)\mathbf{Q}(z)\Big)_s \qquad \forall t, \forall z \in \overset{\circ}{\Omega}. \quad (6.4)$$

The discretization is carried out on an equidistant grid with stepsize k in time direction and stepsize h in space (z) direction. In the grid points the function $\pi_s(t,z)$ is approximated by the discrete function $u^n_{s,j}$ with

$$u^n_{s,j} \approx \pi_s(nk, z^{\min} + jh) \qquad n = 0, \ldots, T, \quad j = 0, \ldots, N. \quad (6.5)$$

and $\mathbf{u}$ is a column vector. The z-dependent functions are discretized as

$$\begin{aligned} \mu_{s,j} &= \mu_s(z^{\min} + jh) \\ \sigma^2_{s,j} &= \sigma^2_s(z^{\min} + jh) \qquad j = 0, \dots, N \\ q_{s,t,j} &= q_{s,t}(z^{\min} + jh) \end{aligned} \tag{6.6}$$

and $\mathbf{Q}_j = (q_{s,t,j})_{\substack{s=1,\dots,S \\ t=1,\dots,S}}$.

In the following the discretization for the inner grid points will be derived. Then the scheme at the boundary points will be determined. Together they form the system of linear equations that must be solved in each time step. At the end of the section the discretization of the initial conditions is described.

The coupling term is discretized explicitly with using the known time levels n and $(n-1)$ because it does not affect the stability of the scheme (see [Zis98]). The time levels are weighted with the parameter $(1+\tau)$ and $(-\tau)$.

$$\Big(\pi(nk, z^{\min} + jh)\mathbf{Q}\Big)_s \approx (1+\tau)\left(u^n_{0,j}, \dots, u^n_{S,j})\mathbf{Q}_j\right)_s - \tau\left((u^{n-1}_{0,j}, \dots, u^{n-1}_{S,j})\mathbf{Q}_j\right)_s. \tag{6.7}$$

The right hand side of (6.7) is abbreviated $\mathbf{u}_{s,j} \overset{n}{\underset{n-1}{\textcircled{\scriptsize T}}} \mathbf{Q}_j$. The parameter τ is chosen in such a way, that the scheme becomes consistent. In the implementation that has been used for the experiments it has been set equal to θ. In the following the index s for the discrete state will be omitted and an arbitrary state will be used. The time derivative is discretized with a forward difference quotient

$$\frac{\partial}{\partial t}\pi(nk, z^{\min} + jh) \approx \frac{u^{n+1}_j - u^n_j}{k} \tag{6.8}$$

For the first space derivative an upwind parameter ρ is introduced. Hence,

$$\begin{aligned} \frac{\partial}{\partial z}\pi(nk, z^{\min}+jh)\cdot\mu(z^{min}+jh) \approx{}& \theta\left((1-\rho)\frac{u^{n+1}_j\mu_j - u^{n+1}_{j-1}\mu_{j-1}}{h} + \rho\frac{u^{n+1}_{j+1}\mu_{j+1} - u^{n+1}_j\mu_j}{h}\right) \\ &+ (1-\theta)\left((1-\rho)\frac{u^n_j\mu_j - u^n_{j-1}\mu_{j-1}}{h} + \rho\frac{u^n_{j+1}\mu_{j+1} - u^n_j\mu_j}{h}\right) \end{aligned} \tag{6.9}$$

where for

$$\mu_j > 0 \Rightarrow 0 < \rho_j < \frac{1}{2} \qquad \text{the left bound difference weighs more, and for} \tag{6.10a}$$

$$\mu_j < 0 \Rightarrow \frac{1}{2} < \rho_j < 1 \qquad \text{the right bound difference weighs more, and} \tag{6.10b}$$

$\rho = \frac{1}{2}$ is the symmetric difference. In the implementation ρ is computed as

$$\rho = \frac{\sigma^2_j}{2h\mu_j} \qquad j = 0, \dots, N.$$

If $\mu_j > 0$ then ρ is bounded by 0.5 from above and if $\mu_j < 0$ then ρ is bounded by 0.5 from below. For $\mu_j = 0$ it is set to 0.5.

The second derivative is discretized with a central difference

$$\frac{\partial^2}{\partial z^2}\pi(nk, z^{\min} + jh) \cdot \sigma^2(z^{min} + jh) \approx \theta \frac{u_{j-1}^{n+1}\sigma_{j-1}^2 - 2u_j^{n+1}\sigma_j^2 + u_{j+1}^{n+1}\sigma_{j+1}^2}{h^2} + (1-\theta)\frac{u_{j-1}^{n}\sigma_{j-1}^2 - 2u_j^{n}\sigma_j^2 + u_{j+1}^{n}\sigma_{j+1}^2}{h^2}. \quad (6.11)$$

Then the discrete approximation of (6.1) is

$$\begin{aligned}\frac{u_j^{n+1} - u_j^n}{k} &+ \theta\left((1-\rho)\frac{u_j^{n+1}\mu_j - u_{j-1}^{n+1}\mu_{j-1}}{h} + \rho\frac{u_{j+1}^{n+1}\mu_{j+1} - u_j^{n+1}\mu_j}{h}\right) \\ &+ (1-\theta)\left((1-\rho)\frac{u_j^{n}\mu_j - u_{j-1}^{n}\mu_{j-1}}{h} + \rho\frac{u_{j+1}^{n}\mu_{j+1} - u_j^{n}\mu_j}{h}\right) \\ &= \frac{1}{2}\theta\frac{u_{j-1}^{n+1}\sigma_{j-1}^2 - 2u_j^{n+1}\sigma_j^2 + u_{j+1}^{n+1}\sigma_{j+1}^2}{h^2} \\ &\quad + \frac{1}{2}(1-\theta)\frac{u_{j-1}^{n}\sigma_{j-1}^2 - 2u_j^{n}\sigma_j^2 + u_{j+1}^{n}\sigma_{j+1}^2}{h^2} + \mathbf{u}_j \underset{n-1}{\overset{n}{\textcircled{\scriptsize T}}} \mathbf{Q}_j \quad (6.12)\end{aligned}$$

Ordered by the grid points and with $r = \frac{k}{h^2}$ this becomes

$$\begin{aligned}u_{j-1}^{n+1}(-\theta r(h(1-\rho)\mu_{j-1} - \frac{1}{2}\sigma_{j-1}^2)) + u_j^{n+1}(1 + \theta r(h(1-\rho)\mu_j - h\rho\mu_j + \sigma^2)) &+ \\ u_{j+1}^{n+1}(\theta r(h\rho\mu_{j+1} - \frac{1}{2}\sigma_{j+1}^2)) = u_{j-1}^{n}((1-\theta)r(h(1-\rho)\mu_{j-1} + \frac{1}{2}\sigma_{j-1}^2)) &+ \\ u_j^n(1 + (1-\theta)r(-h(1-\rho)\mu_j + h\rho\mu_j - \sigma^2)) &+ \\ u_{j+1}^n((1-\theta)r(-h\rho\mu_{j+1} + \frac{1}{2}\sigma_{j+1}^2)) + \mathbf{u}_j \underset{n-1}{\overset{n}{\textcircled{\scriptsize T}}} \mathbf{Q}_j & \quad (6.13)\end{aligned}$$

Using the abbreviations $a_j = h(1-\rho)\mu_j + \frac{1}{2}\sigma_j^2$ and $b_j = -h\rho\mu_j + \frac{1}{2}\sigma_j$ further simplifies the equation and the final scheme is

$$\begin{aligned}u_{j-1}^{n+1}(-\theta r a_{j-1}) &+ u_j^{n+1}(1 + \theta r(a_j + b_j)) + u_{j+1}^{n+1}(-\theta r b_{j+1}) \\ &= u_{j-1}^n((1-\theta)r a_{j-1}) + u_j^n(1 - (1-\theta)r(a_j + b_j)) \\ &\quad + u_{j+1}^n((1-\theta)r b_{j+1}) + \mathbf{u}_j \underset{n-1}{\overset{n}{\textcircled{\scriptsize T}}} \mathbf{Q}_j \\ &\quad \text{for } j = 1, \ldots, N-1, n = 0, \ldots, T-1. \quad (6.14)\end{aligned}$$

The boundary conditions can not be obtained by simply discretizing the boundary condition (5.52), since this does not preserve probability mass. Instead the discrete boundary conditions were derived through the fact that the density integrates to one, which is in consistence with (5.52) as has been shown in [Zis98]. The conservation of probability mass is formalized as

$$\sum_{s=1}^{S} \int_{\Omega} \pi_s(t,z) = 1 \tag{6.15}$$

which takes on the discrete form

$$h \sum_{s=1}^{S} \sum_{j=0}^{N} u_{s,j}^{n} = 1 \qquad \forall n = 0, \dots, T. \tag{6.16}$$

The discrete scheme (6.14) has to be summed over all inner grid points and the use of the discrete mass conservation is used as an additional condition. A condition for the boundary points is obtained which assures that finally the sum over the scheme for all grid points equals one.

Both sides of (6.14) are summed over all discrete states $s = 1, \dots, S$ and all inner grid points $j = 1, \dots, N-1$ and the indices of summation are transformed so that the terms in the sum only are u_j^n and u_j^{n+1}.

$$\begin{aligned}
\sum_{s=1}^{S} \left(\sum_{j=0}^{N-2} u_{s,j}^{n+1}(-\theta r a_{s,j}) + \sum_{j=1}^{N-1} u_{s,j}^{n+1}(1 + \theta r(a_{s,j} + b_{s,j})) + \sum_{j=2}^{N} u_{s,j}^{n+1}(-\theta r b_{s,j}) \right) \\
= \sum_{s=1}^{S} \Bigg(\sum_{j=0}^{N-2} u_{s,j}^{n}((1-\theta) r a_{s,j}) + \sum_{j=1}^{N-1} u_{s,j}^{n}(1 - (1-\theta) r(a_{s,j} + b_{s,j})) + \\
\sum_{j=2}^{N} u_{s,j}^{n+1}((1-\theta) r b_{s,j}) + \sum_{j=1}^{N-1} \mathbf{u}_{s,j} \overset{n}{\underset{n-1}{\textcircled{\scriptsize T}}} \mathbf{Q}_j \Bigg)
\end{aligned} \tag{6.17}$$

The summation over the coupling term will be regarded first. The rows of the generator matrix always sum to zero, so

$$\begin{aligned}
\sum_{s=1}^{S} \sum_{j=1}^{N-1} \mathbf{u}_{s,j} \overset{n}{\underset{n-1}{\textcircled{\scriptsize T}}} \mathbf{Q}_j &= \sum_{s=1}^{S} \sum_{j=1}^{N-1} \left((1+\tau) \sum_{l=1}^{S} u_{l,j}^{n} q_{l,s,j} - \tau \sum_{l=1}^{S} u_{l,j}^{n-1} q_{l,s,j} \right) \\
&= (1+\tau) \sum_{l=1}^{S} \left(\sum_{j=1}^{N-1} u_{l,j}^{n} \underbrace{\sum_{s=1}^{S} q_{l,s,j}}_{=0} \right) - \tau \sum_{l=1}^{S} \left(\sum_{j=1}^{N-1} u_{l,j}^{n-1} \underbrace{\sum_{s=1}^{S} q_{l,s,j}}_{=0} \right) = 0.
\end{aligned} \tag{6.18}$$

This property of the coupling term has been used before in Chapter 5.

The rest of (6.17) is now regarded for one discrete state s and the index for the state is omitted again. To get identical limits in all the sums first and last elements are taken out of the sums starting at $j = 1$ and ending with $j = N - 1$.

$$
\begin{aligned}
&\sum_{j=2}^{N-2} u_j^{n+1}\Big(\underbrace{-\theta r a_j + 1 + \theta r(a_j + b_j) - \theta r b_j}_{=1}\Big) + u_0^{n+1}(-\theta r a_0) + u_1^{n+1}(1 + \theta r b_1) \\
&\qquad\qquad + u_{N-1}^{n+1}(1 + \theta r a_{N-1}) + u_N^{n+1}(-\theta r b_N) \\
&= \sum_{j=2}^{N-2} u_j^{n}\Big(\underbrace{(r(1-\theta)a_j) + (1 - (1-\theta)r(a_j + b_j)) + ((1-\theta)r b_j)}_{=1}\Big) + u_0^n(1-\theta)r a_0 \\
&\qquad + u_1^n(1 - (1-\theta)r b_1) + u_{N-1}^n(1 - (1-\theta)r a_{N-1}) + u_N^n(1-\theta)r b_N
\end{aligned} \tag{6.19}
$$

Since (6.16) can be rewritten as

$$
\sum_{s=1}^{S}\sum_{j=2}^{N-2} u_{s,j}^n = \frac{1}{h} - u_{s,0}^n - u_{s,1}^n - u_{s,N-1}^n - u_{s,N}^n \tag{6.20}
$$

6.19 is transformed into

$$
\begin{aligned}
&\frac{1}{h} + u_0^{n+1}(-\theta r a_0 - 1) + u_1^{n+1}(1 + \theta r b_1 - 1) + u_{N-1}^{n+1}(1 + \theta r a_{N-1} - 1) + u_N^{n+1}(-\theta r b_N - 1) \\
&= \frac{1}{h} + u_0^n((1-\theta)r a_0 - 1) + u_1^n(-(1-\theta)r b_1) + u_{N-1}^n(-(1-\theta)r a_{N-1}) + u_N^n((1-\theta)r b_N - 1)
\end{aligned} \tag{6.21}
$$

A stronger condition is imposed by claiming equality of the terms on either end. This is possible because the terms at the boundary points do not affect each other. For the lower bound the following holds

$$
u_0^{n+1}(1 + \theta r a_0) - u_1^{n+1}(\theta) r b_1) = u_0^n(1 - (1-\theta)r a_0) + u_1^n((1-\theta)r b_1) \tag{6.22a}
$$

and at the upper bound

$$
u_N^{n+1}(1 + \theta r b_N) - u_{N-1}^{n+1}(\theta) r a_{N-1}) = u_N^n(1 - (1-\theta)r b_N) + u_{N-1}^n((1-\theta)r a_{N-1}) \tag{6.22b}
$$

For each time step the following system of linear equations has been obtained

$$
\mathbf{L}\mathbf{u}^{n+1} = \mathbf{R}\mathbf{u}^n + \Big(\mathbf{u}_j \overset{n}{\underset{n-1}{Ⓣ}} \mathbf{Q}_j\Big)^T_{j=0,\dots,N} \tag{6.23}
$$

where the matrices are

$$\mathbf{L} = \begin{pmatrix} 1+\theta r a_0 & -\theta r a_1 & 0 & \cdots & 0 \\ -\theta r a_0 & 1+\theta r(a_1+b_1) & -\theta r b_2 & & \\ \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & 0 & -\theta r a_{N-2} & 1+\theta r(a_{N-1}+b_{N-1}) & -\theta r b_n \\ & & 0 & -\theta r a_{N-1} & 1+\theta r b_N \end{pmatrix} \tag{6.24}$$

and

$$\mathbf{R} = \begin{pmatrix} 1-(1-\theta)ra_0 & (1-\theta)rb_1 & 0 & \cdots & 0 \\ (1-\theta)ra_0 & 1-(1-\theta)r(a_1+b_1) & (1-\theta)rb_2 & & \\ \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & 0 & (1-\theta)ra_{N-2} & 1-(1-\theta)r(a_{N-1}+b_{N-1}) & (1-\theta)rb_n \\ & & 0 & (1-\theta)ra_{N-1} & 1-(1-\theta)rb_N \end{pmatrix} \tag{6.25}$$

and the abbreviations have been

$$a_j = h(1-\rho)\mu_j + \frac{1}{2}\sigma_j^2 \qquad b_j = -h\rho\mu_j + \frac{1}{2}\sigma_j^2 \qquad r = \frac{k}{h^2}. \tag{6.26}$$

The transient solution of a second order FSPN is carried out by solving the system of linear equations (6.23) at each time step. Note, that these equations only hold for second order models with $\sigma^2 \neq 0$.

Finally in this section the initial condition will be discretized. The initial condition consists of a Dirac impulse at the location corresponding to the systems initial configuration in the initial discrete state (see 5.38). The Dirac impulse is approximated by a triangle of width d. The number of grid points over which the initial condition is distributed in each direction can be specified with the parameter *initvar*. Hence, $d = 2h \cdot initvar$. At $z_0 = z^{\min} + ih$ (where i is fix and $i = \frac{z_0 - z^{\min}}{h}$) the triangle function $\Delta(jh)$ is defined as

$$\Delta(jh) = \begin{cases} \Delta_0 \left(1 - \frac{|i-j|}{initvar+1}\right) & \text{if } |i-j| \leq initvar \\ 0 & \text{else} \end{cases} \tag{6.27}$$

The factor Δ_0 is determined in such a way, that

$$\int_\Omega \delta(z)dz = 1 \qquad \text{and} \qquad h\sum_{j=1}^{N} \Delta(jh) = 1. \tag{6.28}$$

If *initvar* is chosen large enough, so that points outside the boundaries need to be used, then the value at these points is distributed equally over the grid points lying inside the domain.

The solution of the initial-boundary value problem (6.1) is given by the discrete initial condition (6.27) and the system of linear equations (6.23), that is filled with the discretization of the inner grid points (6.14) and the boundary points (6.22a) and (6.22b).

The discretization approximates the differential equation with order $o(h,k)$ as has been shown in [Zis98] pp. 33 ff. For $\theta = \frac{1}{2}$ the approximation is of order $o(h,k^2)$. The fully implicit scheme ($\theta = 1$) is unconditionally stable. For $\theta \neq 1$ the scheme is stable under the condition that

$$(1-\theta)\sigma^2 \frac{k}{h^2} \leq 1 \tag{6.29}$$

for constant as well as for variable coefficients.

6.1.2 Jump Transitions for Fluid Places

With the introduction of jump transitions concerning the fluid place, the differential equation, that must be solved is (see (5.11))

$$\begin{aligned}\frac{\partial \boldsymbol{\pi}(t,z)}{\partial t} + \frac{\partial}{\partial z}\Big(\boldsymbol{\pi}(t,z)\mathbf{M}(z)\Big) - \frac{1}{2}\frac{\partial^2}{\partial z^2}\Big(\boldsymbol{\pi}(t,z)\boldsymbol{\Sigma}^2(z)\Big) - \boldsymbol{\pi}(t,z)\Big(\mathbf{Q}(z) - \boldsymbol{\Lambda}(z)\Big) - \\ \sum_{\hat{l}=1}^{\hat{L}} \int_0^z \boldsymbol{\pi}(t,z-y)\mathbf{B}_{\hat{l}}(y)\,dy\,\boldsymbol{\Lambda}_{\hat{l}}(z) + \\ \sum_{\check{l}=1}^{\check{L}} \int_z^{z^{\max}} \boldsymbol{\pi}(t,y)\mathbf{B}_{\check{l}}(z-y)\,dy\,\boldsymbol{\Lambda}_{\check{l}}(z) = 0 \qquad \forall t, \forall z \in \mathring{\Omega} = (z^{\min}, z^{\max}).\end{aligned} \tag{6.30}$$

In addition to the discretization, described in the previous section integrations have to be carried out. The integration is for the uniform and the Dirac jump height distributions over discontinuous and not differentiable functions, only the exponential jump height distribution gives a differentiable function to integrate over - after sufficiently many time steps, so that the solution vector $\mathbf{u}$ is continuous. Initially the solution vector has the form of a triangle, whose width depends on the parameter *initvar* (see (6.27)). It turned out that in all cases the rectangular summation comes closest to the convolution integral, which should evaluate to one in the sum over all grid points.

The additional coefficients, that must be evaluated in the grid points are

$$\begin{aligned}\lambda_{s,j} &= \lambda_s(z^{\min} + jh) \\ b_{s,j} &= b_s(z^{\min} + jh)\end{aligned} \qquad j = 0, \dots, N. \tag{6.31}$$

The addition of jumps changes (6.7) into

$$\mathbf{u}_{s,j} \underset{n-1}{\overset{n}{\textcircled{T}}} \Big(\mathbf{Q} - \mathbf{\Lambda}\Big) = (1+\tau)\left(u^n_{0,j}, \dots, u^n_{S,j})\Big(\mathbf{Q} - \mathbf{\Lambda}\Big)\right)_s - \tau\left((u^{n-1}_{0,j}, \dots, u^{n-1}_{S,j})\Big(\mathbf{Q} - \mathbf{\Lambda}\Big)\right)_s . \tag{6.32}$$

Let v^n_j be the numerical approximation of the integral due to a jump up and $\tilde{v}^n_j$, the numerical approximation of the integral due to a jump down, that is

$$\begin{aligned} v^n_j &\approx \int\limits_0^{jh} \pi(nk, z^{\min} + ih)\ b((j-i)h)\ d(ih) \\ \tilde{v}^n_j &\approx \int\limits_{jh}^{Nh} \pi(nk, z^{\min} + ih)\ b((i-j)h)\ d(ih) \end{aligned} \tag{6.33}$$

The integrals are always computed explicitly only on the time level n. The computation of the integral is carried out separately for each of the three jump height distributions. First, the exponential distribution is applied. For each grid point $j = 0, \dots, N-1$ the following integral is computed for a jump up

$$\int\limits_0^{jh} \pi(nk, z^{\min} + ih)\ \text{Exponential}((j-i)h)d(ih) \approx \frac{1}{2}u^n_0 \cdot \eta + \sum_{i=1}^{j} u^n_i \eta e^{-\eta(j-i)h} = v^n_j \tag{6.34}$$

where η is the parameter of the exponential distribution. For $j = N$ the integral is approximated by

$$v^n_N = \frac{1}{2}u^n_N \cdot \eta + \sum_{i=1}^{N-1} u^n_i \eta e^{-\eta(N-i)h} + \frac{1}{2}u^n_0 \eta e^{-\eta N h}. \tag{6.35}$$

For a jump down the integral is

$$\int\limits_{jh}^{Nh} \pi(nk, z^{\min} + ih)\ \text{Exponential}((i-j)h)d(ih) \approx \frac{1}{2}u^n_j \cdot \eta + \sum_{i=j+1}^{N} u^n_i \eta e^{-\eta(i-j)h} = \tilde{v}^n_j$$

$$j = 0, \dots N-1, \tag{6.36}$$

and

$$\tilde{v}^n_N = \frac{1}{2}u^n_N \cdot \eta. \tag{6.37}$$

If $b = Dirac(lh) = \delta(j-l)\cdot\frac{1}{h}$ there is only one term in each sum unequal to zero for $j = 0, \ldots, N$, hence for a jump up

$$\int\limits_0^{jh} \pi(nk, z^{\min} + ih)\ \mathrm{Dirac}((j-i)h)dy \approx \sum_{i=0}^{j} u_i^n \cdot \delta((j-i)-l)\frac{1}{h} = u_{l-j}^n \cdot \frac{1}{h} = v_j^n \qquad j \leq l \tag{6.38}$$

otherwise $v_j^n = 0$. For a jump down

$$\int\limits_{jh}^{Nh} \pi(nk, z^{\min} + ih)\ \mathrm{Dirac}(|j-i|h)dy \approx \sum_{i=0}^{j} u_i^n \cdot \delta(i-j-l)\frac{1}{h} = u_{j+l}^n \cdot \frac{1}{h} = \tilde{v}_j^n \tag{6.39}$$

The third jump height distribution is the uniform distribution.

$$\mathrm{Uniform}(i) = \begin{cases} \frac{1}{bh-ah} & a < i \leq b \\ 0 & else \end{cases}$$

which gives the following approximation to the integral due to a jump up

$$\begin{aligned}&\int_0^{jh} \pi(nk, z^{\min} + ih)\cdot\ \mathrm{Uniform}((j-i)h)dy \\ &\qquad\approx \sum_{i=0}^{j} u_i^n \frac{1}{bh-ah}(\delta(a+1-(j-i)) + \ldots + \delta(b-(j-i))) = v_j^n. \end{aligned}\tag{6.40}$$

For a jump down the integral evaluates to

$$\begin{aligned}&\int_{jh}^{Nh} \pi(nk, z^{\min} + ih)\cdot\ \mathrm{Uniform}((i-j)h)dy \\ &\qquad\approx \sum_{i=0}^{j} u_i^n \frac{1}{bh-ah}(\delta(a+1-(i-j)) + \ldots + \delta(b-(i-j))) = \tilde{v}_j^n. \end{aligned}\tag{6.41}$$

Now at last the probability mass that gets lost because the jumps happen on a finite axis has to be cared about. The reasoning will be done only for jumps up. For the jumps down the analogous arguments hold. The amount of probability mass, that goes beyond the upper bound can be quantified to be:

$$\begin{aligned}1 - h\sum_{j=0}^{N}\int_0^{jh} \pi(nk, z^{\min} + (j-i)h)\ b(ih)\ d(ih) &\approx 1 - h\sum_{j=0}^{N}\sum_{i=0}^{j} u_{j-i}^n b_i \\ = h\sum_{j=N+1}^{\infty}\sum_{i=0}^{N} u_{N-i}^n b((j-N+i)\cdot h) &= h\sum_{j=0}^{\infty}\sum_{i=0}^{N} u_{N-i}^n b((j+i)\cdot h)\end{aligned}\tag{6.42}$$

So, if the integration terms for all grid points are added, the amount this sum is less than one is the amount of probability mass, that has gone beyond the boundary. The first term from the left in (6.42) describes that quantity. In the second term the integral is replaced by its discrete approximation when the rectangular rule is used. Would there be an infinite number of grid points (i.e. the axis is unbounded to the right and $N = \infty$) the double summation would evaluate to one (with some numerical error) and (6.42) would equal zero. The amount the double sum is less than one is computed regarding the cut-off terms $j = N+1, \dots, \infty$. This is done in the third term in (6.42). The last term is obtained by transformation of the summation index, so that the starting point is $j = 0$. In other words, what is missing in the end is the sum over the probability vector times the probability of doing a jump large enough to go across the boundary. If $B(.)$ is the probability distribution function and $\overline{B}(.)$ the survival function corresponding to $b(.)$, then the amount that is missing is

$$h \sum_{i=0}^{N} u_{N-i}^{n} \sum_{j=0}^{\infty} b((i+j)\cdot h) = h \sum_{i=0}^{N} u_{N-i}^{n} P(jump > ih), \tag{6.43}$$

which is equivalent to

$$\text{rest} = h \sum_{i=0}^{N} u_{N-i}^{n} \overline{B}(i \cdot h). \tag{6.44}$$

Formula (6.44) is used for the two different boundary treatments. If the original situation shall be preserved in case no jump inside the range is possible, then (6.44) is used to determine for each u_j the amount that has been taken away with the jump integrals but could not be placed else where, since the grid point it should have been taken to does not ly inside the regarded domain. So, for the strategy *preserve* the terms that are sought are

$$w_{N-i}^{n} = v_{N-i}^{n} \Big(1 + \overline{B}(ih)\Big)_{i=0,\dots,N}. \tag{6.45}$$

For a jump down the equivalent is

$$\tilde{w}_{i}^{n} = \tilde{v}_{i}^{n} \Big(1 + \overline{B}(ih)\Big)_{i=0,\dots,N}. \tag{6.46}$$

If on the other hand in any case a jump as large as possible shall be carried out, then the strategy is called *force_jump* and

$$\begin{array}{rclrcl} w_{N}^{n} &=& v_{N}^{n}\Big(1+\frac{1}{2}\text{rest}\Big) & w_{N-1}^{n} &=& v_{N-1}^{n}\Big(1+\frac{1}{4}\text{rest}\Big) \\ w_{N-2}^{n} &=& v_{N-2}^{n}\Big(1+\frac{1}{6}\text{rest}\Big) & w_{N-3}^{n} &=& v_{N-3}^{n}\Big(1+\frac{1}{12}\text{rest}\Big) \\ w_{j}^{n} &=& v_{j}^{n}, & & & j=0,\dots,N-4. \end{array} \tag{6.47}$$

For a jump down the final integration terms are

$$\begin{array}{rclrcl} \tilde{w}_0^n & = & \tilde{v}_0^n\Big(1+\frac{1}{2}\mathrm{rest}\Big) & \tilde{w}_1^n & = & \tilde{v}_1^n\Big(1+\frac{1}{4}\mathrm{rest}\Big) \\ \tilde{w}_2^n & = & \tilde{v}_2^n\Big(1+\frac{1}{6}\mathrm{rest}\Big) & \tilde{w}_3^n & = & \tilde{v}_3^n\Big(1+\frac{1}{12}\mathrm{rest}\Big) \\ \tilde{w}_j^n & = & \tilde{v}_j^n, & & & j=0,\dots,N-4. \end{array} \tag{6.48}$$

The system of linear equations that must be solved in each time-step is

$$\mathbf{Lu}^{n+1} = \mathbf{Ru}^n + \Big(\mathbf{u}_j \overset{n}{\underset{n-1}{\textcircled{\scriptsize T}}} (\mathbf{Q}-\mathbf{\Lambda})^T_{j=0,\dots,N}\Big) + \mathbf{\Lambda}\Big(\mathbf{W}+\tilde{\mathbf{W}}\Big) \tag{6.49}$$

where $\mathbf{L}$ and $\mathbf{R}$ have been defined in (6.24) and (6.25), $\tilde{\mathbf{W}} = \mathrm{diag}(w_1^n,\dots,w_N^n)$ and $\tilde{\mathbf{W}} = \mathrm{diag}(\tilde{w}_1^n,\dots,\tilde{w}_N^n)$.

6.1.3 The Discretization Scheme in two Continuous Dimensions

The differential equations in two continuous dimensions describe the dynamics of an FSPN with two fluid places. Both dimensions (i.e. fluid places) must have a limited domain $[z_1^{\min}, z_1^{\max}]$, and $[z_2^{\min}, z_2^{\max}]$ respectively. In the interior of that domain the following differential equation holds

$$\begin{aligned} &\frac{\partial \boldsymbol{\pi}(t,z_1,z_2)}{\partial t} + \frac{\partial}{\partial z_1}\Big(\boldsymbol{\pi}(t,z_1,z_2)\mathbf{M}_1(z_1)\Big) + \frac{\partial}{\partial z_2}\Big(\boldsymbol{\pi}(t,z_1,z_2)\mathbf{M}_2(z_2)\Big) \\ &\quad = \frac{1}{2}\frac{\partial^2}{\partial z_1^2}\Big(\boldsymbol{\pi}(t,z_1,z_2)\mathbf{\Sigma}_1^2(z_1)\Big) + \frac{1}{2}\frac{\partial^2}{\partial z_2^2}\Big(\boldsymbol{\pi}(t,z_1,z_2)\mathbf{\Sigma}_2^2(z_2)\Big) + \boldsymbol{\pi}(t,z_1,z_2)\mathbf{Q}(z_1,z_2) \\ &\qquad\qquad \forall t, \forall z \in \mathring{\Omega} = (z_1^{\min}, z_1^{\max}) \times (z_2^{\min}, z_2^{\max}). \end{aligned} \tag{6.50}$$

subject to the boundary condition

$$\begin{aligned} \Big[&\frac{1}{2}\frac{\partial}{\partial z_1}\boldsymbol{\pi}(t,z_1,z_2)\mathbf{\Sigma}_1^2(z_1) + \frac{1}{2}\frac{\partial}{\partial z_2}\boldsymbol{\pi}(t,z_1,z_2)\mathbf{\Sigma}_2^2(z_2) - \boldsymbol{\pi}(t,z_1,z_2)\mathbf{M}_1(z_1) \\ &- \boldsymbol{\pi}(t,z_1,z_2)\mathbf{M}_2(z_2)\Big]_{\mathbf{z}=\{z_1^{\min},z_1^{\max}\},\{z_2^{\min},z_2^{\max}\}} \end{aligned} \tag{6.51}$$

and the initial condition

$$\boldsymbol{\pi}(0,z_1,z_2) = \boldsymbol{\pi}_0\delta(\mathbf{z}-\mathbf{z}_0). \tag{6.52}$$

The discretization is carried out on a rectangular domain of size $(z_1^{\max}-z_1^{\min})\cdot(z_2^{\max}-z_2^{\min})$ with stepsize h_1 in direction of z_1, stepsize h_2 in direction of z_2 and stepsize k in time. Figure 6.1 shows the numbering of the grid points in the two directions in space.

For one state s the probability vector $\pi_s(t, z_1, z_2)$ is approximated by the discrete function u with

$$u^n_{s,j} \approx \pi_s(nk, z_1^{\min} + l_1 h_1, z_2^{\min} + l_2 h_2) \quad \text{for} \quad \begin{aligned} n &= 0, \dots, T \\ l_1 &= 0, \dots, M-1 \\ l_2 &= 0, \dots, N-1 \\ j &= l_2 M + l_1. \end{aligned} \tag{6.53}$$

There are M grid points in direction z_1 with $M = \frac{z_1^{\max} - z_1^{\min}}{h_1} + 1$ and N grid points in direction z_2 with $N = \frac{z_2^{\max} - z_2^{\min}}{h_2} + 1$. Note, that the numbering is not identical to the one-dimensional case, where $N = \frac{z^{\max} - z^{\min}}{h}$ and the number of grid points was N.

The functions μ_k and σ_k^2 and the components $q_{s,t}$ of the generator matrix $\mathbf{Q}$ are evaluated in the grid points to

$$\begin{aligned} \mu_{s,k,j} &= \mu_{s,k}(z_1^{\min} + l_1 h_1, z_2^{\min} + l_2 h_2) \\ \sigma^2_{s,k,j} &= \sigma^2_{s,k}(z_1^{\min} + l_1 h_1, z_2^{\min} + l_2 h_2) \\ q_{s,t,j} &= q_{s,t}(z_1^{\min} + l_1 h_1, z_2^{\min} + l_2 h_2) \end{aligned} \qquad \begin{aligned} l_1 &= 0, \dots, M-1 \\ l_2 &= 0, \dots, N-1 \\ k &= 1, 2 \\ j &= l_2 M + l_1 \end{aligned} \tag{6.54}$$

and

$$\mathbf{Q}_j = (q_{s,t,j})_{\substack{s=1,\dots,S \\ t=1,\dots,S}}.$$

The coupling term is defined as in the one-dimensional case

$$\left(\mathbf{u}_j \underset{n-1}{\overset{n}{Ⓣ}} \mathbf{Q}_j\right)_s = (1+\tau)\Big((u^n_{0,j}, \dots, u^n_{S,j})\mathbf{Q}_j\Big)_s - \tau\Big((u^{n-1}_{0,j}, \dots, u^{n-1}_{S,j})\mathbf{Q}_j\Big)_s. \tag{6.55}$$

The equation is discretized with using the five points u_j, u_{j-1}, u_{j+1}, u_{j+M}, and u_{j-M}. As in the discretization of the one-dimensional equation this leads to a system of linear equations for the inner grid points. The factors of the boundary points are determined by summation over the scheme and the use of the discretized mass conservation law. The corner points are not touched by the five point formulation for the inner grid points and hence they will neither come into play through the calculation of the boundary points. But since the corner points will be of interest in the solutions, they are later determined by extrapolation. Finally the whole scheme has to be normalized.

The time derivative is discretized explicitly, for the first derivative in space the upwind parameter τ and the parameter for implicity θ are used. The second derivative in space

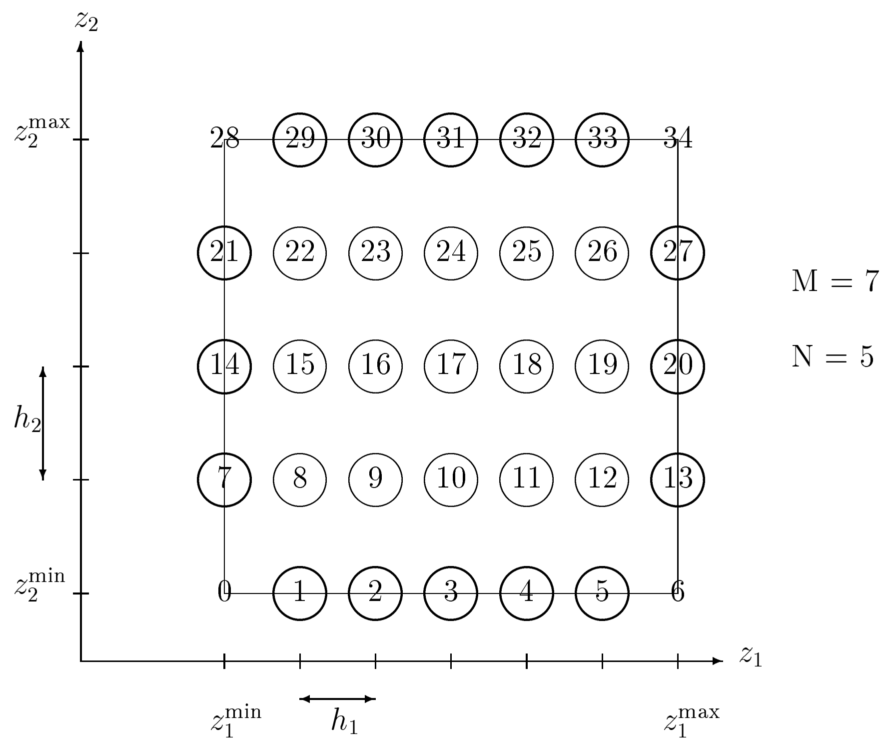

Figure 6.1: Numbering of the grid points

only requires θ. The full scheme is for one discrete state (without the index s)

$$
\begin{aligned}
\frac{u_j^{n+1}-u_j^n}{k} &+ \theta r_1\left(\rho\frac{u_{j+1}^{n+1}\mu_{1,j+1}-u_j^{n+1}\mu_{1,j}}{h_1}+(1-\rho)\frac{u_j^{n+1}\mu_{1,j}-u_{j-1}^{n+1}\mu_{1,j-1}}{h_1}\right)\\
&+(1-\theta)r_2\left(\rho\frac{u_{j+1}^{n}\mu_{1,j+1}-u_j^{n}\mu_{1,j}}{h_1}+(1-\rho)\frac{u_j^{n}\mu_{1,j}-u_{j-1}^{n}\mu_{1,j-1}}{h_1}\right)\\
&+\theta r_1\left(\rho\frac{u_{j+M}^{n+1}\mu_{2,j+M}-u_j^{n+1}\mu_{2,j}}{h_2}+(1-\rho)\frac{u_j^{n+1}\mu_{2,j}-u_{j-M}^{n+1}\mu_{2,j-M}}{h_2}\right)\\
&+(1-\theta)r_2\left(\rho\frac{u_{j+M}^{n}\mu_{2,j+M}-u_j^{n}\mu_{2,j}}{h_2}+(1-\rho)\frac{u_j^{n}\mu_{2,j}-u_{j-M}^{n}\mu_{2,j-M}}{h_2}\right)\\
= \quad &\frac{1}{2}\theta r_1\left(\frac{u_{j+1}^{n+1}\sigma_{1,j+1}^2-2u_j^{n+1}\sigma_{1,j}^2+u_{j-1}^{n+1}\sigma_{1,j-1}^2}{h_1^2}\right)\\
&+\frac{1}{2}(1-\theta)r_2\left(\frac{u_{j+1}^{n}\sigma_{1,j+1}^2-2u_j^{n}\sigma_{1,j}^2+u_{j-1}^{n}\sigma_{1,j-1}^2}{h_1^2}\right)\\
&+\frac{1}{2}\theta r_1\left(\frac{u_{j+M}^{n+1}\sigma_{2,j+M}^2-2u_j^{n+1}\sigma_{2,j}^2+u_{j-M}^{n+1}\sigma_{2,j-M}^2}{h_2^2}\right)\\
&+\frac{1}{2}(1-\theta)r_2\left(\frac{u_{j+M}^{n}\sigma_{2,j+M}^2-2u_j^{n}\sigma_{2,j}^2+u_{j-M}^{n}\sigma_{2,j-M}^2}{h_2^2}\right)+\mathbf{u}_j\underset{n-1}{\overset{n}{ⓣ}}\mathbf{Q}_j.
\end{aligned}
\tag{6.56}
$$

Ordered by the grid points and with $r_1 = \frac{k}{h_1^2}$ and $r_2 = \frac{k}{h_2^2}$ the scheme becomes

$$\begin{aligned}
&u_{j-M}^{n+1}\Big(\theta r_2\big(-(1-\rho)h_2\mu_{2,j-M} - \frac{1}{2}\sigma_{2,j-M}^2\big)\Big) + u_{j-1}^{n+1}\Big(\theta r_1\big(-(1-\rho)h_1\mu_{1,j-1} - \frac{1}{2}\sigma_{1,j-1}^2\big)\Big)\\
&+u_j^{n+1}\bigg(1+\theta\Big(r_1\big((1-2\rho)h_1\mu_{1,j}+\sigma_{1,j}^2\big) + r_2\big((1-2\rho)h_2\mu_{2,j}+\sigma_{2,j}^2\big)\Big)\bigg)\\
&+u_{j+1}^{n+1}\Big(\theta r_1\big(\rho h_1\mu_{1,j+1} - \frac{1}{2}\sigma_{1,j+1}^2\big)\Big) + u_{j+M}^{n+1}\Big(\theta r_2\big(\rho h_2\mu_{2,j+M} - \frac{1}{2}\sigma_{2,j+M}^2\big)\Big)\\
&= u_{j-M}^{n}\bigg((1-\theta)r_2\Big((1-\rho)h_2\mu_{2,j-M} + \frac{1}{2}\sigma_{2,j-M}^2\Big)\bigg) + u_{j-1}^{n}\bigg((1-\theta)r_1\Big((1-\rho)h_1\mu_{1,j-1} + \frac{1}{2}\sigma_{1,j-1}^2\Big)\bigg)\\
&+u_j^{n}\Big(1+(1-\theta)\big(r_1\big((2\rho-1)h_1\mu_{1,j}-\sigma_{1,j}^2\big) + r_2\big((2\rho-1)h_2\mu_{2,j}-\sigma_{2,j}^2\big)\big)\Big)\\
&+u_{j+1}^{n}\Big((1-\theta)r_1\big(-\rho h_1\mu_{1,j+1} + \frac{1}{2}\sigma_{1,j+1}^2\big)\Big) + u_{j+M}^{n}\Big((1-\theta)r_2\big(-\rho h_2\mu_{2,j+M} + \frac{1}{2}\sigma_{2,j+M}^2\big)\Big)\\
&+\mathbf{u}_j \overset{n}{\underset{n-1}{\textcircled{\scriptsize T}}} \mathbf{Q}_j.
\end{aligned} \tag{6.57}$$

Using the abbreviations

$$a_{i,j} = r_i\Big((1-\rho)h_i\mu_{i,j} + \frac{1}{2}\sigma_{i,j}^2\Big) \qquad b_{i,j} = r_i\Big(-\rho h_i\mu_{i,j} + \frac{1}{2}\sigma_{i,j}^2\Big) \tag{6.58}$$

the final scheme on the inner grid points is

$$\begin{aligned}
&u_{j-M}^{n+1}(-\theta a_{2,j-M}) + u_{j-1}^{n+1}(-\theta a_{1,j-1}) + u_j^{n+1}\left(1+\theta(a_{1,j}+b_{1,j}+a_{2,j}+b_{2,j})\right)\\
&\quad+u_{j+1}^{n+1}(-\theta b_{1,j+1}) + u_{j+M}^{n+1}(-\theta b_{2,j+M})\\
&\quad=u_{j-M}^{n}((1-\theta)a_{2,j-M}) + u_{j-1}^{n}((1-\theta)a_{1,j-M}) +\\
&\quad u_j^{n}(1+(1-\theta)(-a_{1,j}-b_{1,j}-a_{2,j}-b_{2,j})) + u_{j+1}^{n}((1-\theta)b_{1,j+1})\\
&\quad+u_{j+M}^{n}((1-\theta)b_{2,j+M}) + \mathbf{u}_j \overset{n}{\underset{n-1}{\textcircled{\scriptsize T}}} \mathbf{Q}_j\\
&\qquad n=0,\dots,T, j=M+1,\dots,(N-1)M-2.
\end{aligned} \tag{6.59}$$

As in the one-dimensional case the scheme for the boundary points cannot be obtained by simply discretizing the boundary condition. Instead, here as well the scheme over the inner grid points has to be summed and with using a discrete form of the probability mass conservation law (6.60) conditions for the boundary points are derived.

The two-dimensional probability mass conservation law is

$$\sum_{s=1}^{S} \int_{z_1^{\min}}^{z_1^{\max}} \int_{z_2^{\min}}^{z_2^{\max}} \pi_s(t, z_1, z_2)\, dz_1\, dz_2 = 1 \tag{6.60}$$

It is discretized with assuming now, that each grid point lies in the center of a cell of size $h_1 \cdot h_2$ and the surface of the two-dimensional probability density in this rectangle is approximated with $u_j \cdot h_1 \cdot h_2$. Hence, it follows for the discrete approximation of (6.60)

$$h_1 h_2 \sum_{s=1}^{S} \sum_{l_2=0}^{N-1} \sum_{l_1=0}^{M-1} u_{s,l_2 M+l_1} = 1. \tag{6.61}$$

When using the sum in the above form, the corner points are taken into account as well. Replacing u_{s,l_2M+l_1} with the discrete scheme in all inner points leads to equations for the corner points, that have no unique solution. Justified by the fact that the corner points are not used in the discretization scheme of the differential equation they are also excluded from the sum (6.60) giving the following sum that can be divided in sums for the inner grid points and the boundaries as shown in figure 6.1.

$$\sum_{s=1}^{S} h_1 h_2 \Big(\underbrace{\sum_{l_2=1}^{N-2} \sum_{l_1=1}^{M-2} u_{s,l_2M+l_1}}_{\text{inner grid points}} + \underbrace{\sum_{l_1=1}^{M-2} u_{s,l_1}}_{\text{lower bound}} + \underbrace{\sum_{l_1=1}^{M-2} u_{s,(N-1)M+l_1}}_{\text{upper bound}} + \underbrace{\sum_{l_2=1}^{N-2} u_{s,l_2M}}_{\text{left side}} + \underbrace{\sum_{l_2=1}^{N-2} u_{s,(l_2+1)M-1}}_{\text{right side}} \Big) = 1 \tag{6.62}$$

The scheme for the inner grid points is now summed up. Note, that the indices for the numbering is in terms of l_1 and l_2, so that $u_j = u_{l_2M+l_1}$, $u_{j-M} = u_{(l_2-1)M+l_1}$, etc.

$$\begin{aligned}
&\sum_{s=1}^{S} \sum_{l_2=1}^{N-2} \sum_{l_1=1}^{M-2} \Bigg(u^{n+1}_{s,(l_2-1)M+l_1} \left(-\theta a_{2,(l_2-1)M+l_1}\right) + u^{n+1}_{s,l_2M+l_1-1} \left(-\theta a_{1,l_2M+l_1-1}\right) \\
&\quad + u^{n+1}_{s,l_2M+l_1} \left(1 + \theta \left(a_{1,l_2M+l_1} + b_{1,l_2M+l_1} + a_{2,l_2M+l_1} + b_{2,l_2M+l_1}\right)\right) \\
&\quad + u^{n+1}_{s,l_2M+l_1+1} \left(-\theta b_{1,l_2M+l_1+1}\right) + u^{n+1}_{s,(l_2+1)M+l_1} \left(-\theta b_{2,(l_2+1)M+l_1}\right) \Bigg) \\
= &\sum_{s=1}^{S} \sum_{l_2=1}^{N-2} \sum_{l_1=1}^{M-2} \Bigg(u^{n}_{s,(l_2-1)M+l_1} \left((1-\theta) a_{2,(l_2-1)M+l_1}\right) + u^{n}_{s,l_2M+l_1-1} \left((1-\theta) a_{1,l_2M+l_1-1}\right) \\
&\quad + u^{n}_{s,l_2M+l_1} \Big(1 + (1-\theta)(-a_{1,l_2M+l_1} - b_{1,l_2M+l_1} - a_{2,l_2M+l_1} - b_{2,l_2M+l_1})\Big) \\
&\quad + u^{n}_{s,l_2M+l_1+1} \left((1-\theta) b_{1,l_2M+l_1+1}\right) + u^{n}_{s,(l_2+1)M+l_1} \left((1-\theta) b_{2,(l_2+1)M+l_1}\right) + \Big(\mathbf{u}_{l_2M+l_1} \overset{n}{\underset{n-1}{Ⓣ}} \mathbf{Q}_j \Big)_s \Bigg)
\end{aligned} \tag{6.63}$$

The coupling term is

$$\sum_{s=1}^{S}\sum_{l_2=1}^{N-2}\sum_{l_1=1}^{M-2}\mathbf{u}_{s,l_2M+l_1}\overset{n}{\underset{n-1}{\textcircled{\tau}}}\mathbf{Q}_{l_2M+l_1}=\sum_{s=1}^{S}\sum_{l_2=1}^{N-2}\sum_{l_1=1}^{M-2}\Bigg((1+\tau)\sum_{j=1}^{S}u^{n}_{j,l_2m+l_1}q_{j,s,l_2M+l_1}-\tau\sum_{j=1}^{S}u^{n-1}_{j,l_2M+l_1}q_{j,s,l_2M+l_1}\Bigg)=0. \quad (6.64)$$

after the same transformations as in the one-dimensional case. So, the coupling term is no longer present and for the rest of the equation only one discrete state is regarded currently and, as usual, the index s is omitted. The indices of the sums are transformed and first or last terms are taken out of the sums, so that only $u^{n}_{l_2M+l_1}$ and $u^{n+1}_{l_2M+l_1}$ are used in the summations

$$
\begin{aligned}
&\sum_{l_2=1}^{N-2}\sum_{l_1=1}^{M-2} u^{n+1}_{l_2M+l_1}\Bigg(1+\theta\Big(\underbrace{-a_{2,l_2M+l_1}-a_{1,l_2M+l_1}+a_{2,l_2M+l_1}+a_{1,l_2M+l_1}}_{=0}\\
&\quad+\underbrace{b_{2,l_2M+l_1}+b_{1,l_2M+l_1}-b_{2,l_2M+l_1}-b_{1,l_2M+l_1}}_{=0}\Big)\Bigg)\\
&\quad+\sum_{l_1=1}^{M-2} u^{n+1}_{l_1}(-\theta a_{2,l_1})-\sum_{l_1=1}^{M-2} u^{n+1}_{(N-2)M+l_1}(-\theta a_{2,(N-2)M+l_1})\\
&\quad+\sum_{l_2=1}^{N-2} u^{n+1}_{l_2M}(-\theta a_{1,l_2M})-\sum_{l_2=2}^{N-1} u^{n+1}_{l_2M-2}(-\theta a_{1,l_2M-2})\\
&\quad+\sum_{l_2=2}^{N-1} u^{n+1}_{l_2M-1}(-\theta b_{1,l_2M-1})-\sum_{l_2=1}^{N-2} u^{n+1}_{l_2M+1}(-\theta b_{1,l_2M+1})\\
&\quad+\sum_{l_1=1}^{M-2} u^{n+1}_{(N-1)M+l_1}(-\theta b_{2,(N-1)M+l_1})-\sum_{l_1=1}^{M-2} u^{n+1}_{M+l_1}(-\theta b_{2,M+l_1})\\
=\;&\sum_{l_2=1}^{N-2}\sum_{l_1=1}^{M-2} u^{n}_{l_2M+l_1}\Bigg(1+(1-\theta)\Big(\underbrace{a_{2,l_2M+l_1}+a_{1,l_2M+l_1}-a_{1,l_2M+l_1}+a_{2,l_2M+l_1}}_{=0}\\
&\quad+\underbrace{-b_{1,l_2M+l_1}-b_{2,l_2M+l_1}+b_{1,l_2M+l_1}+b_{2,l_2M+l_1}}_{=0}\Big)\Bigg)\\
&\quad+\sum_{l_1=1}^{M-2} u^{n}_{l_1}((1-\theta)a_{2,l_1})-\sum_{l_1=1}^{M-2} u^{n}_{(N-2)M+l_1}((1-\theta)a_{2,(N-2)M+l_1})\\
&\quad+\sum_{l_2=1}^{N-2} u^{n}_{l_2M}((1-\theta)a_{1,l_2M})-\sum_{l_2=2}^{N-1} u^{n}_{l_2M-2}((1-\theta)a_{1,l_2M-2})\\
&\quad+\sum_{l_2=2}^{N-1} u^{n}_{l_2M-1}((1-\theta)b_{1,l_2M-1})-\sum_{l_2=1}^{N-2} u^{n}_{l_2M+1}((1-\theta)b_{1,l_2M+1})\\
&\quad+\sum_{l_1=1}^{M-2} u^{n}_{(N-1)M+l_1}((1-\theta)b_{2,(N-1)M+l_1})-\sum_{l_1=1}^{M-2} u^{n}_{M+l_1}((1-\theta)b_{2,M+l_1}).
\end{aligned}
\tag{6.65}
$$

Ordering by sums over l_1 and l_2 gives

$$
\begin{aligned}
&\sum_{l_2=1}^{N-2}\sum_{l_1=1}^{M-2} u^{n+1}_{l_2M+l_1} + \theta \sum_{l_1=1}^{M-2}\Big(- u^{n+1}_{l_1} a_{2,l_1} + u^{n+1}_{(N-2)M+l_1} a_{2,(N-2)M+l_1} \\
&\qquad -u^{n+1}_{(N-1)M+l_1} b_{2,(N-1)M+l_1} + u^{n+1}_{M+l_1} b_{2,M+l_1}\Big) \\
&\qquad +\theta \sum_{l_2=1}^{N-2}\Big(- u^{n+1}_{l_2M} a_{1,l_2M} + u^{n+1}_{(l_2+1)M-2} a_{1,(l_2+1)M-2} \\
&\qquad -u^{n+1}_{(l_2+1)M-1} b_{1,(l_2+1)M-1} + u^{n+1}_{l_2M+1} b_{1,l_2M+1}\Big) \\
&\quad = \sum_{l_2=1}^{N-2}\sum_{l_1=1}^{M-2} u^{n}_{l_2M+l_1} + (1-\theta) \sum_{l_1=1}^{M-2}\Big(- u^{n}_{l_1} a_{2,l_1} - u^{n}_{(N-2)M+l_1} a_{2,(N-2)M+l_1} \\
&\qquad +u^{n}_{(N-1)M+l_1} b_{2,(N-1)M+l_1} - u^{n}_{M+l_1} b_{2,M+l_1}\Big) \\
&\qquad +(1-\theta) \sum_{l_2=1}^{N-2}\Big(u^{n}_{l_2M} a_{1,l_2M} - u^{n}_{(l_2+1)M-2} a_{1,(l_2+1)M-2} \\
&\qquad +u^{n}_{(l_2+1)M-1} b_{1,(l_2+1)M-1} - u^{n}_{l_2M+1} b_{1,l_2M+1}\Big).
\end{aligned}
\tag{6.66}
$$

The discrete law of conservation of probability mass (6.62) is now transformed into

$$
\begin{aligned}
\sum_{s=1}^{S}\sum_{l_2=1}^{N-2}\sum_{l_1=1}^{M-2} u_{s,l_2M+l_1} \quad = \quad & \frac{1}{h_1h_2} - \sum_{s=1}^{S}\sum_{l_1=1}^{M-2} u_{s,l_1} - \sum_{s=1}^{S}\sum_{l_1=1}^{M-2} u_{s,(N-1)M+l_1} - \sum_{s=1}^{S}\sum_{l_2=1}^{N-2} u_{s,l_2M} \\
& - \sum_{s=1}^{S}\sum_{l_2=1}^{N-2} u_{s,(l_2+1)M-1}.
\end{aligned}
\tag{6.67}
$$

This holds on both time levels n and $n+1$, so in (6.66) the sum over the inner grid points can be replaced. After collecting the terms for each grid point in each discrete state and then showing only one arbitrary discrete state the following holds

$$
\begin{aligned}
&\sum_{l_1=1}^{M-2}\Big(- u_{l_1}^{n+1}\big(\theta a_{2,l_1}+1\big) + u_{(N-2)M+l_1}^{n+1}\theta a_{2,(N-2)M+l_1} - u_{(N-1)M+l_1}^{n+1}\big(\theta b_{2,(N-1)M+l_1}+1\big)\\
&\quad + u_{M+l_1}^{n+1}\theta b_{2,M+l_1}\Big) + \sum_{l_2=1}^{N-2}\Big(u_{l_2M}^{n+1}\big(-\theta a_{1,l_2M}-1\big) + u_{(l_2+1)M-2}^{n+1}\theta a_{1,(l_2+1)M-2}\\
&\quad + u_{(l_2+1)M-1}^{n+1}\big(-\theta b_{1,(l_2+1)M-1}-1\big) + u_{l_2M+1}^{n+1}\theta b_{1,l_2M+1}\Big)\\
&\quad = \sum_{l_1=1}^{M-2}\Big(u_{l_1}^{n}\big((1-\theta)a_{2,l_1}-1\big) - u_{(N-2)M+l_1}^{n}(1-\theta)a_{2,(N-2)M+l_1}\\
&\quad + u_{(N-1)M+l_1}^{n}\big((1-\theta)b_{2,(N-1)M+l_1}-1\big) - u_{M+l_1}^{n}(1-\theta)b_{2,M+l_1}\Big) + \sum_{l_2=1}^{N-2}\Big(u_{l_2M}^{n}\big((1-\theta)a_{1,l_2M}-1\big)\\
&\quad - u_{(l_2+1)M-2}^{n}(1-\theta)a_{1,(l_2+1)M-2} + u_{(l_2+1)M-1}^{n}\big((1-\theta)b_{1,(l_2+1)M-1}-1\big) - u_{l_2M+1}^{n}(1-\theta)b_{1,l_2M+1}\Big).
\end{aligned}
\tag{6.68}
$$

This equation can now be split into four equations, one belonging to each boundary. Then the sums are split further, so that equality of the single terms is claimed. In doing so, the conditions imposed on the boundary points are stronger and (6.68) is fulfilled. On the lower bound holds

$$
\Big(u_{l_1}^{n+1}\big(1-\theta a_{2,l_1}\big) + u_{M+l_1}^{n+1}\theta b_{2,M+l_1}\Big) = \Big(u_{l_1}^{n}\big(1-(1-\theta)a_{2,l_1}\big) + u_{M+l_1}^{n}(1-\theta)b_{2,M+l_1}\Big)
$$

$$
l_1 = 1,\dots,M-2. \tag{6.69a}
$$

On the upper bound holds

$$
\begin{aligned}
&\Big(u_{(N-1)M+l_1}^{n+1}\big(\theta b_{2,(N-1)M+l_1}+1\big) - u_{(N-2)M+l_1}^{n+1}\theta a_{2,(N-2)M+l_1}\Big)\\
&\qquad = \Big(+u_{(N-1)M+l_1}^{n}\big((1-\theta)b_{2,(N-1)M+l_1}-1\big) - u_{(N-2)M+l_1}^{n}(1-\theta)a_{2,(N-2)M+l_1}\Big)
\end{aligned}
$$

$$
l_1 = 1,\dots,M-2. \tag{6.69b}
$$

On the left side the boundary condition is

$$
\begin{aligned}
&\Big(u_{l_2M}^{n+1}\big(\theta a_{1,l_2M}+1\big) - u_{l_2M+1}^{n+1}\theta b_{1,l_2M+1}\Big)\\
&\qquad = \Big(u_{l_2M}^{n}\big(1-(1-\theta)a_{1,l_2M}\big) + u_{l_2M+1}^{n}(1-\theta)b_{1,l_2M+1}\Big)
\end{aligned}
$$

$$
l_2 = 1,\dots,N-2. \tag{6.69c}
$$

And on the right side

$$\Big(u^{n+1}_{(l_2+1)M-1}\big(\theta b_{1,(l_2+1)M-1}+1\big)-u^{n+1}_{(l_2+1)M-2}\theta a_{1,(l_2+1)M-2}\Big)$$
$$=\Big(u^{n}_{(l_2+1)M-1}\big(1-(1-\theta)b_{1,(l_2+1)M-1}\big)+u^{n}_{(l_2+1)M-2}(1-\theta)a_{1,(l_2+1)M-2}\Big)$$
$$l_2=1,\dots,N-2. \quad (6.69d)$$

The system of linear equations that is obtained for each discrete state again is of the form

$$\mathbf{L}\mathbf{u}^{n+1}=\mathbf{R}\mathbf{u}^{n}+\Big(\mathbf{u}_j \overset{n}{\underset{n-1}{\textcircled{\scriptsize T}}} \mathbf{Q}\Big)^T_{j=0,\dots,NM-1} \tag{6.70}$$

with

$$\mathbf{u}^n=\begin{pmatrix} u_0^n \\ \vdots \\ u^n_{NM-1}\end{pmatrix}$$

and $\mathbf{u}^{n+1}$ analogously.

The matrices $\mathbf{L}$ and $\mathbf{R}$ are no tridiagonal matrices any longer. They have dimension $NM\times NM$ and consist of $N\times$ blocks of size $M\times M$. The blocks are either zero matrices, diagonal or tridiagonal ones, with

$$\mathbf{L}=\begin{pmatrix} \mathbf{B}_l & \mathbf{B}_{+l} & & & \mathbf{0} \\ \mathbf{D}_{-1} & \mathbf{T}_1 & \mathbf{D}_{+1} & & \\ & \ddots & \ddots & \ddots & \\ & \ddots & \mathbf{D}_{-(N-2)} & \mathbf{T}_{N-2} & \mathbf{D}_{+(N-2)} \\ \mathbf{0} & & & \mathbf{B}_{-u} & \mathbf{B}_u \end{pmatrix} \tag{6.71}$$

and

$$\mathbf{R}=\begin{pmatrix} \tilde{\mathbf{B}}_l & \tilde{\mathbf{B}}_{+l} & & & \mathbf{0} \\ \tilde{\mathbf{D}}_{-1} & \tilde{\mathbf{T}}_1 & \tilde{\mathbf{D}}_{+1} & & \\ & \ddots & \ddots & \ddots & \\ & \ddots & \tilde{\mathbf{D}}_{-(N-2)} & \tilde{\mathbf{T}}_{N-2} & \tilde{\mathbf{D}}_{+(N-2)} \\ \mathbf{0} & & & \tilde{\mathbf{B}}_{-u} & \tilde{\mathbf{B}}_u \end{pmatrix} \tag{6.72}$$

The matrices $\mathbf{B}_l, \mathbf{B}_{+l}$, $\mathbf{B}_{-u}$ and $\mathbf{B}_u$ contain the boundary condition for the lower bound and the upper bound respectively. The points in grid row i are represented by the matrices $\mathbf{D}_{-i}, \mathbf{T}_i$ and $\mathbf{D}_{+i}$, where $\mathbf{D}_{\pm i}$ denote the diagonal matrices and $\mathbf{T}_i$ a tridiagonal matrix.

The first and last row of the matrix $\mathbf{T}_i$ holds the left and right boundary condition on row i. The analogous holds for the matrices constructing $\mathbf{R}$ with the tildes. The entries of the matrices are the following

$$\mathbf{B}_l = \operatorname{diag}\begin{pmatrix} 1 \\ 1-\theta a_{2,1} \\ \vdots \\ 1-\theta a_{2,M-2} \\ 1 \end{pmatrix} \qquad \mathbf{B}_{+l} = \theta\cdot\operatorname{diag}\begin{pmatrix} 0 \\ b_{2,M+1} \\ \vdots \\ b_{2,2M-2} \\ 0 \end{pmatrix}$$

$$\mathbf{B}_{-\mathbf{u}} = \theta\cdot\operatorname{diag}\begin{pmatrix} 0 \\ -a_{2,(N-2)M+1} \\ \vdots \\ -a_{2,(N-1)M-2} \\ 0 \end{pmatrix} \qquad \mathbf{B}_u = \operatorname{diag}\begin{pmatrix} 1 \\ \theta b_{2,(N-1)M+1}+1 \\ \vdots \\ \theta b_{2,NM-2}+1 \\ 1 \end{pmatrix} \tag{6.73}$$

$$\mathbf{D}_{-\mathbf{i}} = \theta\cdot\operatorname{diag}\begin{pmatrix} 0 \\ -a_{2,(i-1)M+1} \\ \vdots \\ -a_{2,iM-2} \\ 0 \end{pmatrix} \qquad \mathbf{D}_{+i} = \theta\cdot\operatorname{diag}\begin{pmatrix} 0 \\ -b_{2,(i+1)M+1} \\ \vdots \\ -b_{2,(i+2)M-2} \\ 0 \end{pmatrix}$$

$$\mathbf{T}_i = \begin{pmatrix} 1+\theta a_{1,iM} & -\theta b_{1,iM+1} & & & 0 \\ -\theta a_{1,iM} & 1+\theta c_{iM+1} & \theta b_{1,iM+2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\theta a_{1,(i+1)M-3} & 1+\theta c_{(i+1)M-2} & -\theta b_{1,(i+1)M-1} \\ 0 & & & -\theta a_{1,(i+1)M-2} & 1+\theta b_{1,(i+1)M-1} \end{pmatrix} \tag{6.74}$$

where $c_l = a_{1,l} + b_{1,l} + a_{2,l} + b_{2,l}$. The blocks in the matrix $\mathbf{R}$ are

$$\tilde{\mathbf{B}}_l = \operatorname{diag}\begin{pmatrix} 1 \\ 1-(1-\theta)a_{2,1} \\ \vdots \\ 1-(1-\theta)a_{2,M-2} \\ 1 \end{pmatrix} \qquad \tilde{\mathbf{B}}_{+l} = (1-\theta)\cdot\operatorname{diag}\begin{pmatrix} 0 \\ b_{2,M+1} \\ \vdots \\ b_{2,2M-2} \\ 0 \end{pmatrix}$$

$$\tilde{\mathbf{B}}_{-u} = (1-\theta)\cdot\mathrm{diag}\begin{pmatrix} 0 \\ -a_{2,(N-2)M+1} \\ \vdots \\ -a_{2,(N-1)M-2} \\ 0 \end{pmatrix} \qquad \tilde{\mathbf{B}}_{u} = \mathrm{diag}\begin{pmatrix} 1 \\ (1-\theta)b_{2,(N-1)M+1}-1 \\ \vdots \\ (1-\theta)b_{2,NM-2}-1 \\ 1 \end{pmatrix} \tag{6.75}$$

$$\tilde{\mathbf{D}}_{-i} = (1-\theta)\mathrm{diag}\begin{pmatrix} 0 \\ a_{2,(i-1)M+1} \\ \vdots \\ a_{2,iM-2} \\ 0 \end{pmatrix} \qquad \tilde{\mathbf{D}}_{+i} = (1-\theta)\mathrm{diag}\begin{pmatrix} 0 \\ b_{2,(i+1)M+1} \\ \vdots \\ b_{2,(i+2)M-2} \\ 0 \end{pmatrix}$$

$$\tilde{\mathbf{T}}_i = \begin{pmatrix} 1+(1-\theta)a_{1,iM} & (1-\theta)b_{1,iM+1} & & & 0 \\ (1-\theta)a_{1,iM} & 1-(1-\theta)c_{iM+1} & \theta b_{1,iM+2} & & \\ & \ddots & \ddots & \ddots & \\ & & (1-\theta)a_{1,(i+1)M-3} & 1-(1-\theta)c_{(i+1)M-2} & (1-\theta)b_{1,(i+1)M-1} \\ 0 & & & (1-\theta)a_{1,(i+1)M-2} & 1-(1-\theta)b_{1,(i+1)M-1} \end{pmatrix}. \tag{6.76}$$

The initial conditions are defined similarly to the one-dimensional ones. Instead of a triangle here a pyramid is used to approximate (6.52). Let $\mathbf{z}_0 = \binom{ih_1}{jh_2}$ with i and j fix. Then the pyramid function is defined as

$$\Delta\begin{pmatrix} ih_1 \\ jh_2 \end{pmatrix} = \begin{cases} \Delta_0\cdot\left(1-\frac{|i-l_1|+|j-l_2|}{2\cdot initvar+1}\right) & \text{if } |i-l_1|,|j-l_2|\le initvar \\ 0 & \text{else} \end{cases} \qquad \begin{matrix} \forall l_1 = 0,\dots,M-1, \\ \forall l_2 = 0.\dots,N-1. \end{matrix} \tag{6.77}$$

The factor Δ_0 is determined through the normalization

$$h_1h_2\sum_{l_1=0}^{M-1}\sum_{l_2=0}^{N-1}\Delta\begin{pmatrix} l_1h_1 \\ l_2h_2 \end{pmatrix} = 1. \tag{6.78}$$

Since the coupling term is discretized explicitly, the discrete states can be gone through one by one. This has the advantage of using little memory because the matrices for only one discrete state must be stored at a time. It has on the other hand the disadvantage of requiring much computation, that could be avoided. The matrices **L** and **R** are constant throughout the solution procedure, but they have to be computed again in every time step.

It becomes clear, that the transient solution of second order FSPNs lends itself extremely well to parallelization. One iteration in time can be performed for all discrete states in parallel, followed by a new evaluation of the coupling term. The discretization matrices should, if possible, be kept in memory. In consequence, they need be computed only once in the beginning of the solution process. This would surely have an enormous impact on the run time, but it requires a new, parallel implementation of the discretization scheme.

The jumps have not yet been transferred to the two-dimensional models. Even in one continuous dimension accuracy is severely disturbed by the integrations and it is not clear, whether two-dimensional integration is tractable.

At this point the transient solution of the second order FSPN is complete. The corner points are not used at all in the solution procedure. But since they are of interest for computing measures and also for the graphical representation of the obtained density, they shall be extrapolated with a second order polynomial using the neighbouring grid points. The extrapolating function that is fitted is

$$u(z_1, z_2) = az_1^2 + bz_2^2 + cz_1z_2 + dz_1 + ez_2 + f. \tag{6.79}$$

To determine the six unknowns a through f, six known grid points are needed. The two neighbouring points both in direction of z_1 and z_2 and on the diagonal are used. These are $u(h_1, 0)$, $u(2h_1, 0)$, $u(0, h_2)$, $u(0, 2h_2)$, $u(h_1, h_2)$, and $u(2h_1, 2h_2)$.

Evaluating the polynomial in these points gives

$$\begin{array}{lrcl}
\text{I} & u_1 & = & ah_1^2 + dh_1 + f \\
\text{II} & u_2 & = & 4ah_1^2 + 2dh_1 + f \\
\text{III} & u_M & = & bh_2^2 + eh_2 + f \\
\text{IV} & u_{2M} & = & 4bh_2^2 + 2eh_2 + f \\
\text{V} & u_{M+1} & = & ah_1^2 + bh_2^2 + ch_1h_2 + dh_1 + eh_2 + f \\
\text{VI} & u_{2M+2} & = & 4ah_1^2 + 4bh_2^2 + 4ch_1h_2 + 2dh_1 + 2eh_2 + f \\
 & & & \\
\text{VII: VI - 4 V} & u_{2M+2} - 4u_{M+1} & = & -2dh_1 - 2eh_2 - 3f \\
\text{VIII: II - 4 I} & u_2 - 4u_1 & = & -2dh_1 - 3f \\
\text{IX: IV - 4 III} & u_{2M} - 4u_M & = & -2eh_2 - 3f \\
 & & & \\
\text{X: VII - VIII - IX} & 3f & = & u_{2M+2} - 4u_{M+1} + 4u_1 - u_2 + 4u_M - u_{2M}
\end{array}$$

Hence, it follows for the lower left corner $u(0,0) = f$

$$u_0 = \frac{1}{3}\big(4u_1 - u_2 + 4u_M - 4u_{M+1} - u_{2M} + u_{2M+2}\big) \tag{6.80a}$$

and for the other corners analogously. The lower right corner evaluates to

$$u_{M-1} = \frac{1}{3}\big(4u_{M-2} - u_{M-3} + 4u_{2M-1} - 4u_{2M-2} - u_{3M-1} + u_{3M-3}\big) \tag{6.80b}$$

The upper left corner evaluates to

$$u_{(N-1)M} = \frac{1}{3}\big(4u_{(N-1)M+1} - u_{(N-1)M+2} + 4u_{(N-2)M} - 4u_{(N-2)M+1} - u_{(N-3)M} + u_{(N-3)M+2}\big) \tag{6.80c}$$

and the upper right corner evaluates to

$$u_{NM-1} = \frac{1}{3}\big(4u_{NM-2} - u_{NM-3} + 4u_{(N-1)M-1} - 4u_{(N-1)M-2} - u_{(N-2)M-1} + u_{(N-2)M-3}\big). \tag{6.80d}$$

6.2 Discretization Methods for the Stationary Analysis

As it is common in the analysis of stochastic processes, the stationary equations underlying a Markov process with mixed continuous and discrete state space are simpler than the transient ones. But for these processes a new kind of problem arises. In the stationary analysis the coupling term cannot be evaluated in each time step and hence the states can not be handled one by one, implying that they have to be discretized all in one. So the complexity of the system of linear equations that must be solved grows by the factor of the number of discrete states. In consequence this solution method is not applicable to large models. The splitting into subproblems, one for each discrete state, can be done by applying and alternating direction implicit (ADI) scheme [Str89] which is less accurate. An accurate but slower way of carrying out stationary analysis has been chosen to be implemented first.

6.2.1 The Discretization Scheme for one Continuous Dimension

The equation that must be solved for the stationary analysis of second order FSPN models with one fluid place is

$$\frac{\partial}{\partial z}\Big(\boldsymbol{\pi}(z)\mathbf{M}(z)\Big) = \frac{1}{2}\frac{\partial^2}{\partial z^2}\Big(\boldsymbol{\pi}(z)\boldsymbol{\Sigma}^2(z)\Big) + \boldsymbol{\pi}(z)\mathbf{Q}(z) \qquad \forall z \in \mathring{\Omega} = (z^{\min}, z^{\max}). \tag{6.81}$$

subject to the boundary condition

$$\left[\frac{1}{2}\frac{\partial}{\partial z}\boldsymbol{\pi}(z)\Sigma^2(z) - \boldsymbol{\pi}(z)\mathbf{M}(z)\right]_{z=\{z^{\min},z^{\max}\}}. \tag{6.82}$$

The problem (6.81) and (6.82) however, has no unique solution. Therefore, an additional equation is needed, the probability mass conservation law. It corresponds to the condition in GSPNs that the sum of all state probabilities must equal one. The additional equation is

$$\sum_{s=1}^{S}\int_{z^{\min}}^{z^{\max}} \pi_s(z)\,dz = 1. \tag{6.83}$$

As for the transient analysis the discretization is carried out on an equidistant grid in one dimension with stepsize h. In the grid-points the function $\pi(z)$ is approximated by the discrete function u with

$$u_{s,j} \approx \pi_s(z^{\min} + jh) \qquad j = 0, \ldots, N. \tag{6.84}$$

The z-dependent functions are discretized as defined in (6.6). The coupling term is approximated

$$\big(\boldsymbol{\pi}(z^{\min}+jh)\mathbf{Q}(z^{\min}+jh)\big)_s \approx ((u_{0,j},\ldots,u_{S,j})\mathbf{Q}_j)_s \tag{6.85}$$

the right side of the equation being abbreviated $(\mathbf{u}_j\mathbf{Q}_j)_s$.

The discrete probability mass conservation law is

$$h\sum_{s=1}^{S}\sum_{j=0}^{N} u_{s,j} = 1. \tag{6.86}$$

In the inner grid points the first derivative in z is discretized with an upwind scheme with parameter ρ and the second derivative is discretized with a central difference. The discrete form of (6.81) is for one discrete state s

$$(1-\rho)\frac{u_j\mu_j - u_{j-1}\mu_{j-1}}{h} + \rho\frac{u_{j+1}\mu_{j+1} - u_j\mu_j}{h} = \frac{1}{2}\frac{u_{j-1}\sigma_{j-1}^2 - 2u_j\sigma_j^2 + u_{j+1}\sigma_{j+1}^2}{h^2} + (\mathbf{u}_j\mathbf{Q}_j)_s \tag{6.87}$$

where the index s is omitted in all but the coupling terms and ρ is as described in (6.10a).

Sorted by the grid points and with $r = \frac{1}{h^2}$ equation (6.87) becomes

$$\begin{aligned} 0 = u_{j-1}\left(\frac{1}{2}r\sigma_{j-1}^2 + (1-\rho)rh\mu_{j-1}\right) + u_j\big(-r\sigma_j^2 + \rho rh\mu_j - (1-\rho)rh\mu_j\big) \\ + u_{j+1}\left(\frac{1}{2}r\sigma_{j+1}^2 - \rho rh\mu_{j+1}\right) + (\mathbf{u}_j\mathbf{Q}_j)_s. \end{aligned} \tag{6.88}$$

Using the abbreviations

$$a_j = h(1-\rho)\mu_j + \frac{1}{2}\sigma_j^2 \qquad \text{and} \qquad b_j = -\rho h \mu_j + \frac{1}{2}\sigma_j^2 \tag{6.89}$$

the above equation simplifies to the final discretization scheme for the inner grid points

$$0 = u_{j-1}\left(ra_{j-1}\right) + u_j\left(-r(a_j + b_j)\right) + u_{j+1}\left(rb_{j+1}\right) + \left(\mathbf{u}_j\mathbf{Q}_j\right)_s. \tag{6.90}$$

The scheme in the boundary points is obtained by summation over all inner grid points. In the inner grid points (6.90) as well as (6.86), so the sum is over both equations. The sum over all inner grid points in all states evaluates to

$$\sum_{s=1}^{S}\sum_{j=1}^{N-1} u_{s,j-1}\left(ra_{j-1}\right) + u_{s,j}\left(-r(a2_j + b_j)\right) + u_{s,j+1}\left(rb_{j+1}\right) + \left(\mathbf{u}_j\mathbf{Q}_j\right)_s + u_{s,j}$$
$$= \frac{1}{h} - u_0 - u_N. \tag{6.91}$$

The term

$$\sum_{s=1}^{S}\sum_{j=1}^{N-1} \left(\mathbf{u}_j\mathbf{Q}_j\right)_s = 0 \tag{6.92}$$

can be dropped, as shown in Section 6.1.1.

The indices are transformed and first and last terms of the sum are taken out, so that only u_j still is present

$$\sum_{s=1}^{S}\sum_{j=2}^{N-2} u_{s,j}r\left(a_j - a_j - b_j + b_j + 1\right) + u_{s,0}ra_0 + u_{s,1}ra_1 - u_{s,1}r(a_1 + b_1)$$
$$- u_{N-1}r(a_{N-1} + b_{N-1}) + u_{s,N-1}rb_{N-1} + u_{s,N}rb_N = \frac{1}{h} - u_0 - u_N \tag{6.93}$$

The former equation is equivalent to

$$\sum_{s=1}^{S}\sum_{j=2}^{N-2} u_{s,j} + u_{s,0}ra_0 - u_{s,1}rb_1 - u_{s,N-1}ra_{N-1} + u_{s,N}rb_N = \frac{1}{h} - u_0 - u_N \tag{6.94}$$

and (6.86) can be rewritten as

$$\sum_{s=1}^{S}\sum_{j=2}^{N-2} us, j = \frac{1}{h} - u_0 - u_1 - u_{N-1} - u_N. \tag{6.95}$$

With (6.95) the sum in (6.94) can be replaced, giving

$$-u_1 - u_{N-1} + u_{s,0} r a_0 - u_{s,1} r b_1 - u_{s,N-1} r a_{N-1} + u_{s,N} r b_N = 0 \tag{6.96}$$

This equation is split into two, one for the lower bound and one for the upper boundary in each discrete state s

$$u_1(1 + r b_1) - u_0 r a_0 = 0 \tag{6.97a}$$
$$u_{N-1}(1 + r a_{N-1}) - u_N r b_N = 0 \tag{6.97b}$$

For the discrete scheme in the boundary points the consistency with the theoretical boundary condition shall be proven. The abbreviations r, a_j and b_j are resubstituted in (6.97a), giving

$$u_1 \left(1 - \frac{\rho\mu_1}{h} + \frac{1}{2}\frac{\sigma_1^2}{h^2}\right) - u_0 \left(\frac{(1-\rho)\mu_0}{h} + \frac{1}{2}\frac{\sigma_0^2}{h^2}\right) = 0. \tag{6.98}$$

The above equation is multiplied with h and the terms are sorted by the powers of h

$$h u_1 - (\rho\mu_1 u_1 + (1-\rho)\mu_0 u_0) + \frac{1}{2}\frac{\sigma_1^2 u_1 - \sigma_0^2 u_0}{h} = 0. \tag{6.99}$$

At this place some Taylor series expansions are needed, as there are

$$\begin{array}{lrcl}
\mathrm{I} & \pi(z) & = & \pi(z+h) + o(h) \\
\mathrm{II} : (1-\rho)\pi(z) + \rho\mathrm{I} & \pi(z) & = & (1-\rho)\pi(z) + \rho\pi(z+h) + o(h) \\
\mathrm{III} & \frac{\sigma^2(z+h)\pi(z+h) - \sigma^2(z)\pi(z)}{h} & = & \frac{\partial}{\partial z}\left(\sigma^2(z)\pi(z)\right) + o(h).
\end{array}$$

Herewith the transformation of (6.99) can continue

$$\begin{aligned}
& h u_1 - (\rho\mu_1 u_1 + (1-\rho)\mu_0 u_0) + \frac{1}{2}\frac{\sigma_1^2 u_1 - \sigma_0^2 u_0}{h} \\
&\quad = o(h) - \left(\rho\mu_1 \pi(z^{\min} + h) + (1-\rho)\mu_0 \pi(z^{\min})\right) + \frac{1}{2}\frac{\sigma_1^2 \pi(z^{\min} + h) - \sigma_0^2 \pi(z^{\min})}{h} \\
&\quad \overset{\mathrm{II,III}}{=} o(h) - \mu(z^{\min})\pi(z^{\min}) + o(h) + \frac{1}{2}\frac{\partial}{\partial z^{\min}}\left(\sigma^2(z^{\min})\pi(z^{\min})\right) + o(h) \\
&\quad = \frac{1}{2}\frac{\partial}{\partial z}\left(\sigma^2(z^{\min})\pi(z^{\min})\right) - \mu(z^{\min})\pi(z^{\min}) + o(h) = 0.
\end{aligned} \tag{6.100}$$

Hence, the theoretical boundary condition (6.82) is approximated by the discrete one (6.97a) with order $o(h)$. The consistency of the upper bound discretization follows analogously.

The stationary analysis is carried out by solving the system of equations

$$\mathbf{0} = \mathbf{R}\mathbf{v} + \left((\mathbf{u}_j\mathbf{Q}_j)_s\right)^T_{\substack{j=0,\dots,N\\ s=1,\dots,S}} \tag{6.101}$$

where $\mathbf{v}$ is a column vector of length NS with

$$\mathbf{v} = (u_{1,0}, \dots, u_{1,N}, u_{2,0}, \dots, u_{2,N}, \dots, u_{S,0}, \dots u_{S,N})^T \tag{6.102}$$

and $\mathbf{R}$ is a $NS \times NS$ matrix that consists of S times S blocks of size $N \times N$. On the diagonal the blocks are tridiagonal matrices and all other blocks are zero matrices.

$$\mathbf{R} = \begin{pmatrix} \mathbf{D}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{D}_S \end{pmatrix} \tag{6.103}$$

with

$$\mathbf{D}_s = \begin{pmatrix} -ra_{s,0} & 1 + rb_{s,1} & & & 0 \\ ra_{s,0} & -r(a_{s,1} + b_{s,1}) & rb_{s,2} & & \\ & \ddots & \ddots & \ddots & \\ & & ra_{s,N-2} & -r(a_{s,N-1} + b_{S,N-1}) & rb_{s,N} \\ 0 & & & 1 + ra_{s,N-1} & -rb_{s,N} \end{pmatrix} \tag{6.104}$$

The normalization condition (6.86) has been used to derive the discretization in the boundary points but it has not yet been incorporated in the system of linear equations (6.101). There are different ways of doing so. One is to remove the last $N-1$ rows in the matrix $\mathbf{R}$ and replace the last row with ones. This implies, that the last block $\mathbf{D}_s$ is dropped and instead the normalization condition is added. The zero vector on the left side in (6.101) must be replaced with

$$(\underbrace{0, \dots, 0}_{(S-1)\cdot N elements}, 1). \tag{6.105}$$

The vector $\mathbf{v}$ is replaced with

$$\mathbf{v}' = (u_{1,0}, \dots, u_{1,N}, u_{2,0}, \dots, u_{2,N}, \dots, u_{S-1,0}, \dots u_{S-1,N}, 1)^T \tag{6.106}$$

Hence, the system of linear equations, that must be solved has dimension $N(S-1)+1 \times NS$. This method has the disadvantage, that iterative methods for solving (6.101) get very slow.

But direct solution methods do give correct results. Finally, the probability distribution over the continuous axis in the last discrete state must be determined by solving

$$\mathrm{diag}(\mathbf{1} - \sum_{s=1}^{S-1} \mathbf{u}_s)\mathbf{D}_s \mathbf{v}_s = \mathbf{0}.$$

An alternative method is to cancel out the first column and row in the generator matrix and set the elements of the solution vector belonging to the first discrete state, $u_0 = 1, \ldots, u_N = 1$. Furthermore, the first block $\mathbf{D}_1$ must be set to zero. The reduced system of linear equations, that has only size $N(S-1) \times N(S-1)$ must be solved and subsequently the solution vector u has to be normalized. This method has the advantage, that both iterative and direct solution methods can be used and it still further than the first method reduces the dimension of the system of linear equations to be solved. As when using the other method, finally the probability distribution over the first discrete state is determined by solving

$$\mathrm{diag}(\mathbf{1} - \sum_{s=2}^{S} \mathbf{u}_s)\mathbf{D}_1 \mathbf{v} = \mathbf{0}.$$

6.2.2 The Discretization Scheme in two Continuous Dimensions

To carry out the stationary analysis of a second order FSPN the following differential equation must be solved

$$\begin{aligned}\frac{\partial}{\partial z_1}\Big(\boldsymbol{\pi}(z_1, z_2)\mathbf{M}_1(z_1)\Big) + \frac{\partial}{\partial z_2}\Big(\boldsymbol{\pi}(z_1, z_2)\mathbf{M}_2(z_2)\Big) \\ = \frac{1}{2}\frac{\partial^2}{\partial z_1^2}\Big(\boldsymbol{\pi}(z_1, z_2)\boldsymbol{\Sigma}_1^2(z_1)\Big) + \frac{1}{2}\frac{\partial^2}{\partial z_2^2}\Big(\boldsymbol{\pi}(z_1, z_2)\boldsymbol{\Sigma}_2^2(z_2)\Big) + \boldsymbol{\pi}(z_1, z_2)\mathbf{Q}(z_1, z_2) \\ \forall z \in \mathring{\Omega} = (z_1^{\min}, z_1^{\max}) \times (z_2^{\min}, z_2^{\max}).\end{aligned} \tag{6.107}$$

subject to the boundary condition

$$\begin{aligned}\Big[\frac{1}{2}\frac{\partial}{\partial z_1}\boldsymbol{\pi}(z_1, z_2)\boldsymbol{\Sigma}_1^2(z_1) + \frac{1}{2}\frac{\partial}{\partial z_2}\boldsymbol{\pi}(z_1, z_2)\boldsymbol{\Sigma}_2^2(z_2) - \boldsymbol{\pi}(z_1, z_2)\mathbf{M}_1(z_1) \\ - \boldsymbol{\pi}(z_1, z_2)\mathbf{M}_2(z_2)\Big]_{\mathbf{z}=\{z_1^{\min}, z_1^{\max}\}, \{z_2^{\min}, z_2^{\max}\}}\end{aligned} \tag{6.108}$$

In addition the normalizing condition

$$\sum_{s=1}^{S} \int_{z_1^{\min}}^{z_1^{\max}} \int_{z_2^{\min}}^{z_2^{\max}} \pi_s(z_1, z_2)\, dz_2 dz_1 = 1 \tag{6.109}$$

is needed.

As for the transient analysis the discretization is carried out on a rectangular domain of size $(z_1^{\max} - z_1^{\min}) \cdot (z_2^{\max} - z_2^{\min})$ with stepsize h_1 in direction of z_1 and stepsize h_2 in direction of z_2.

For one state s the probability vector $\pi_s(z_1, z_2)$ is approximated by the discrete function u with

$$u_{s,j} \approx \pi_s(z_1^{\min} + l_1 h_1, z_2^{\min} + l_2 h_2) \quad \text{for} \quad \begin{array}{l} l_1 = 0, \dots, M-1 \\ l_2 = 0, \dots, N-1 \\ j = l_2 M + l_1 \end{array} \tag{6.110}$$

and there are M grid points in direction z_1 with $M = \frac{z_1^{\max} - z_1^{\min}}{h_1} + 1$ and N grid points in direction z_2 with $N = \frac{z_2^{\max} - z_2^{\min}}{h_2} + 1$. The z-dependent functions μ, σ^2 and $\mathbf{Q}$ are defined as in 6.54 and the coupling term as in (6.85). The normalizing condition (6.109) is discretized as

$$h_1 h_2 \sum_{s=1}^{S} \sum_{l_2=0}^{N-1} \sum_{l_1=0}^{M-1} u_{s,l_2 M + l_1} = 1. \tag{6.111}$$

The first derivative in both directions is discretized in an upwind scheme and the second derivative with a central difference. Hence, in direction of z_1 for the approximation of the differential equation u_j the grid points u_{j-1} and u_{j+1} are needed. For the derivatives in direction z_2 the grid points u_{j-M}, u_j and u_{j+M} are used. The complete discrete scheme is the following

$$\begin{aligned} & r_2 \left(\rho \frac{u_{j+1}\mu_{1,j+1} - u_j \mu_{1,j}}{h_1} + (1-\rho) \frac{u_j \mu_{1,j} - u_{j-1}\mu_{1,j-1}}{h_1} \right) \\ & + r_2 \left(\rho \frac{u_{j+M}\mu_{2,j+M} - u_j \mu_{2,j}}{h_2} + (1-\rho) \frac{u_j \mu_{2,j} - u_{j-M}\mu_{2,j-M}}{h_2} \right) \\ = & + \frac{1}{2} r_2 \left(\frac{u_{j+1}\sigma_{1,j+1}^2 - 2u_j \sigma_{1,j}^2 + u_{j-1}\sigma_{1,j-1}^2}{h_1^2} \right) \\ & + \frac{1}{2} r_2 \left(\frac{u_{j+M}^n \sigma_{2,j+M}^2 - 2u_j \sigma_{2,j}^2 + u_{j-M}\sigma_{2,j-M}^2}{h_2^2} \right) + (\mathbf{u}_j \mathbf{Q}_j)_s . \end{aligned} \tag{6.112}$$

Ordered by the grid points and with $r_1 = \frac{1}{h_1^2}$ and $r_2 = \frac{1}{h_2^2}$ the scheme becomes

$$\begin{aligned} 0 = & u_{j-M} \Big(r_2 \Big((1-\rho) h_2 \mu_{2,j-M} + \frac{1}{2}\sigma_{2,j-M}^2 \Big) \Big) + u_{j-1} \Big(r_1 \Big((1-\rho) h_1 \mu_{1,j-1} + \frac{1}{2}\sigma_{1,j-1}^2 \Big) \Big) \\ & + u_j \Big(\Big(r_1 \big((2\rho - 1) h_1 \mu_{1,j} - \sigma_{1,j}^2 \big) + r_2 \big((2\rho - 1) h_2 \mu_{2,j} - \sigma_{2,j}^2 \big) \Big) \Big) \\ & + u_{j+1} \Big(r_1 \big(-\rho h_1 \mu_{1,j+1} + \frac{1}{2}\sigma_{1,j+1}^2 \big) \Big) + u_{j+M} \Big(r_2 \big(-\rho h_2 \mu_{2,j+M} + \frac{1}{2}\sigma_{2,j+M}^2 \big) \Big) + (\mathbf{u}_j \mathbf{Q}_j)_s . \end{aligned} \tag{6.113}$$

Using the abbreviations

$$a_{i,j} = r_i\Big((1-\rho)h_i\mu_{i,j} + \frac{1}{2}\sigma_{i,j}^2\Big) \qquad b_{i,j} = r_i\Big(-\rho h_i\mu_{i,j} + \frac{1}{2}\sigma_{i,j}^2\Big) \tag{6.114}$$

the final scheme on the inner grid points is

$$\begin{aligned} 0 = u_{j-M}\ a_{2,j-M} + u_{j-1}\ a_{1,j-1} + u_j(-a_{1,j} - b_{1,j} - a_{2,j} - b_{2,j}) + u_{j+1}\ b_{1,j+1} \\ + u_{j+M}\ b_{2,j+M} + \big(\mathbf{u}_j\mathbf{Q}_j\big)_s \qquad j = M+1,\dots,(N-1)M-2. \end{aligned} \tag{6.115}$$

The discretization on the four boundaries is determined by summation over the scheme for the inner grid points with using 6.67.

$$\begin{aligned} &\sum_{s=1}^{S}\sum_{l_2=1}^{N-2}\sum_{l_1=1}^{M-2} u_{s,(l_2-1)M+l_1}\ a_{2,(l_2-1)M+l_1} + u_{s,l_2M+l_1-1}\ a_{1,l_2M+l_1-1} \\ &\quad + u_{s,l_2M+l_1}(-a_{1,l_2M+l_1} - b_{1,l_2M+l_1} - a_{2,l_2M+l_1} - b_{2,l_2M+l_1} + 1) + u_{s,l_2M+l_1+1}\ b_{1,l_2M+l_1+1} \\ &\quad + u_{s,(l_2+1)M+l_1}\ b_{2,(l_2+1)M+l_1} + \big(\mathbf{u}_{l_2M+l_1}\mathbf{Q}_{l_2M+l_1}\big)_s \\ &\quad = \frac{1}{h_1h_2} - \sum_{s=1}^{S}\sum_{l_1=1}^{M-2}\big(u_{s,l_1} + u_{s,(N-1)M+l_1}\big) - \sum_{s=1}^{S}\sum_{l_2=1}^{N-2} u_{s,l_2M} + u_{s,(l_2+1)M-1}. \end{aligned} \tag{6.116}$$

As shown in (6.64) for the coupling term holds

$$\sum_{s=1}^{S}\sum_{l_2=1}^{N-2}\sum_{l_1=1}^{M-2}\big(\mathbf{u}_{l_2M+l_1}\mathbf{Q}_{l_2M+l_1}\big)_s = 0. \tag{6.117}$$

Now, the indices are transformed and first or last terms are taken out of the sum, so that only $u_{l_2M+l_1}$ will appear

$$\begin{aligned}
&\sum_{s=1}^{S}\sum_{l_2=1}^{N-2}\sum_{l_1=1}^{M-2} u_{s,l_2M+l_1} \\
&-\sum_{s=1}^{S}\sum_{l_1=1}^{M-2} u_{s,l_1}a_{2,l_1} + \sum_{s=1}^{S}\sum_{l_1=1}^{M-2} u_{s,(N-2)M+l_1}a_{2,(N-2)M+l_1} \\
&-\sum_{s=1}^{S}\sum_{l_2=1}^{N-2} u_{s,l_2M}a_{1,l_2M} + \sum_{s=1}^{S}\sum_{l_2=1}^{N-2} u_{s,l_2M-2}a_{1,l_2M-2} \\
&-\sum_{s=1}^{S}\sum_{l_2=2}^{N-1} u_{s,l_2M-1}b_{1,l_2M-1} + \sum_{s=1}^{S}\sum_{l_2=1}^{N-2} u_{s,l_2M+1}b_{1,l_2M+1} \\
&-\sum_{s=1}^{S}\sum_{l_1=1}^{M-2} u_{s,(N-1)M+l_1}b_{2,(N-1)M+l_1} + \sum_{s=1}^{S}\sum_{l_1=1}^{M-2} u_{s,M+l_1}b_{2,M+l_1} \\
&\quad = \frac{1}{h_1h_2} - \sum_{s=1}^{S}\sum_{l_1=1}^{M-2}\left(u_{s,l_1} + u_{s,(N-1)M+l_1}\right) - \sum_{s=1}^{S}\sum_{l_2=1}^{N-2} u_{s,l_2M} + u_{s,(l_2+1)M-1}.
\end{aligned} \tag{6.118}$$

Another application of 6.67 cancels the triple sum on the left side of the above equation with the right side. Ordering by the sums over l_1 and l_2 gives for each discrete state s

$$\begin{aligned}
&\sum_{l_1=1}^{M-2}(-u_{s,l_1}a_{2,l_1} + u_{s,(N-2)M+l_1}a_{2,(N-2)M+l_1} - u_{s,(N-1)M+l_1}b_{2,(N-1)M+l_1} + u_{s,M+l_1}b_{2,M+l_1}) \\
+&\sum_{l_2=1}^{N-2}(-u_{s,l_2M}a_{1,l_2M} + u_{s,l_2M-2}a_{1,l_2M-2} - u_{s,l_2M-1}b_{1,l_2M-1} + u_{s,l_2M+1}b_{1,l_2M+1} = 0.
\end{aligned} \tag{6.119}$$

This equation can be split into four different equations, one for each boundary. On the lower bound holds

$$u_{s,l_1}a_{2,l_1} - u_{s,M+l_1}b_{2,M+l_1}) \qquad l_1 = 1, \dots, M-2. \tag{6.120a}$$

On the upper bound holds

$$u_{s,(N-1)M+l_1}b_{2,(N-1)M+l_1} - u_{s,(N-2)M+l_1}a_{2,(N-2)M+l_1} \qquad l_1 = 1, \dots, M-2. \tag{6.120b}$$

On the left bound holds

$$u_{s,l_2M}a_{1,l_2M} - u_{s,l_2M+1}b_{1,l_2M+1} \qquad l_2 = 1, \dots, N-2. \tag{6.120c}$$

And on the right bound holds

$$u_{s,l_2M-1}b_{1,l_2M-1} - u_{s,l_2M-2}a_{1,l_2M-2} \qquad l_2 = 1, \dots, N-2. \tag{6.120d}$$

The system of linear equations that must be solved is

$$\mathbf{0} = \mathbf{R}\mathbf{v} + \left((\mathbf{u}_j Q_j)_s\right)^T_{\substack{j=0,\dots,NM-1\\ s=1,\dots,S}} \tag{6.121}$$

where $\mathbf{v}$ is a column vector of length $(NM-1)S$ with

$$\mathbf{v} = (u_{1,0},\dots,u_{1,NM-1},u_{2,0},\dots,u_{2,NM-1},\dots,u_{S,0},\dots u_{S,NM-1})^T \tag{6.122}$$

and $\mathbf{R}$ is a $(NM-1)S \times (NM-1)S$ matrix that consists of S times S blocks of size $NM-1 \times NM-1$. As in the one-dimensional scheme, the blocks on the diagonal have entries and all others are zero

$$\mathbf{R} = \begin{pmatrix} \mathbf{D}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{D}_S \end{pmatrix}. \tag{6.123}$$

The matrices $\mathbf{D}_s$ can be further split into blocks of diagonal and tridiagonal matrices, very similar to the discretization of the transient equation

$$\mathbf{D}_s = \begin{pmatrix} \mathbf{B}_{s,l} & \mathbf{B}_{s,+l} & & & \mathbf{0} \\ \mathbf{D}_{s,-1} & \mathbf{T}_{s,1} & \mathbf{D}_{s,+1} & & \\ & \ddots & \ddots & \ddots & \\ & \ddots & \mathbf{D}_{s,-(N-2)} & \mathbf{T}_{s,N-2} & \mathbf{D}_{s,+(N-2)} \\ \mathbf{0} & & & \mathbf{B}_{s,-u} & \mathbf{B}_{s,u} \end{pmatrix} \tag{6.124}$$

The matrices $\mathbf{B}_{s,l}, \mathbf{B}_{s,+l}$, $\mathbf{B}_{s,-u}$ and $\mathbf{B}_{s,u}$ contain the boundary condition for the lower bound and the upper bound in state s respectively. The points in grid row i in state s are represented by the matrices $\mathbf{D}_{s,-i}, \mathbf{T}_{s,i}$ and $\mathbf{D}_{s,+i}$, where $\mathbf{D}_{s,i}$ denote the diagonal matrices and $\mathbf{T}_{s,i}$ a tridiagonal matrix. The first and last row of the matrix $\mathbf{T}_{s,i}$ holds the left and right boundary condition on row i. The entries of the matrices are the following

$$\mathbf{B}_{s,l} = \operatorname{diag}\begin{pmatrix} 1 \\ -a_{2,1} \\ \vdots \\ -a_{2,M-2} \\ 1 \end{pmatrix} \qquad \mathbf{B}_{s,+l} = \operatorname{diag}\begin{pmatrix} 0 \\ b_{2,M+1} \\ \vdots \\ b_{2,2M-2} \\ 0 \end{pmatrix}$$

$$\mathbf{B}_{s,-u} = \operatorname{diag}\begin{pmatrix} 0 \\ -a_{2,(N-2)M+1} \\ \vdots \\ -a_{2,(N-1)M-2} \\ 0 \end{pmatrix} \qquad \mathbf{B}_{s,u} = \operatorname{diag}\begin{pmatrix} 1 \\ b_{2,(N-1)M+1}+1 \\ \vdots \\ b_{2,NM-2}+1 \\ 1 \end{pmatrix} \tag{6.125}$$

$$\mathbf{D}_{s,-i} = \operatorname{diag}\begin{pmatrix} 0 \\ -a_{2,(i-1)M+1} \\ \vdots \\ -a_{2,iM-2} \\ 0 \end{pmatrix} \qquad \mathbf{D}_{s,+i} = \operatorname{diag}\begin{pmatrix} 0 \\ b_{2,(i+1)M+1} \\ \vdots \\ b_{2,(i+2)M-2} \\ 0 \end{pmatrix}$$

$$\mathbf{T}_{s,i} = \begin{pmatrix} -a_{1,iM} & b_{1,iM+1} & & & 0 \\ a_{1,iM} & 1-c_{iM+1} & b_{1,iM+2} & & \\ & \ddots & \ddots & \ddots & \\ & & a_{1,(i+1)M-3} & 1-c_{(i+1)M-2} & b_{1,(i+1)M-1} \\ 0 & & & -a_{1,(i+1)M-2} & b_{1,(i+1)M-1} \end{pmatrix}. \tag{6.126}$$

where $c_l = a_{1,l} + b_{1,l} + a_{2,l} + b_{2,l}$.

For actually solving the system 6.121 the same two options are available as in the one-dimensional case. The corner points are not computed by the scheme. As in the transient analysis they are extrapolated after the solution on all other points has been obtained.

Chapter 7

Applications

In this chapter the second order FSPN formalism presented in chapter 4 and the numerical solution methods discussed in chapter 6 to solve the equations derived in chapter 5 will be applied to different problems. The purpose in looking at examples is twofold: on the one hand the modelling power of second order FSPNs shall be demonstrated [WH97] and on the other hand the approximation quality as well as the solution components shall be verified.

In the first section the use of second order FSPNs to approximate non-exponentially distributed activities will be discussed [WHG96, WG96]. In the second section accumulated reward will be modelled, which can be done with second order FSPNs explicitly in a straight forward manner by simply using a fluid place to accumulate reward. The mean content of the fluid place represents the mean accumulated reward and the probability density of the fluid variable represents the full probability density of the accumulated reward, which could not be computed with discrete modelling paradigms.

A chemical batch reactor will be modelled in the third section. It is a hybrid system by nature and requires two fluid places in the model, one for the reactor content and the other for its temperature. Hybrid models like the reactor can not be modelled with discrete stochastic Petri nets, unless the systems are simplified severely.

Finally, in the last section of this chapter unfinished work in a queueing system is modelled with second order FSPNs that include jump transitions in the continuous model part. The results are compared with analytical ones, as they are available for simple queueing systems.

All examples used in this chapter are rather small and simple, because for larger second order FSPN models no corresponding discrete model or analytical solution is available and a new modelling paradigm in early stage has to be verified through comparison with known solutions.

The discretization stepsizes in all examples where it is not noted otherwise have been set to $k = 0.01$ in time direction and $h = 0.1$ for the fluid axes. The solutions have all been computed with the fully implicit scheme ($\theta = 1$) and $\tau = 1$.

7.1 Approximation of Non-Exponential Transition Rates

The accuracy of the diffusion approximation for some quite general queueing systems has been studied exhaustively in [RK74]. For queueing systems an analytical formula can be obtained for the diffusion approximation. This closed form solution for the approximate queue length distribution has been used to compute closed form approximations for mean and variance of the queue length, and for the server utilization, which have then been compared with either exact or simulation results.

Two factors influence the approximation quality: the utilization factor ρ and the squared coefficient of variation of the service time distribution. The accuracy of the approximated mean queue size improves as the utilization factor ρ approaches one and the closer the squared coefficient of variation of the service time distribution comes to one.

It has been found out that the diffusion approximation gives a much better approximation to a queueing system with non-exponentially distributed service or inter arrival times than the exponential distribution. Using the exponential distribution for analytical simplicity in cases where it is known not to be suitable is a very common but little accurate alternative to diffusion approximation.

What remains to be shown is, that these results also hold in the context of Petri nets, that is in stochastic processes with more than one discrete state. It will, however, not be expected, that the use of more than one discrete state has a significant impact on the approximation quality. Nevertheless, it has not always been possible to recompute known analytical measures with the numerical algorithm, which is partly due to the fact that stationary measures have been computed with long term transient analysis.

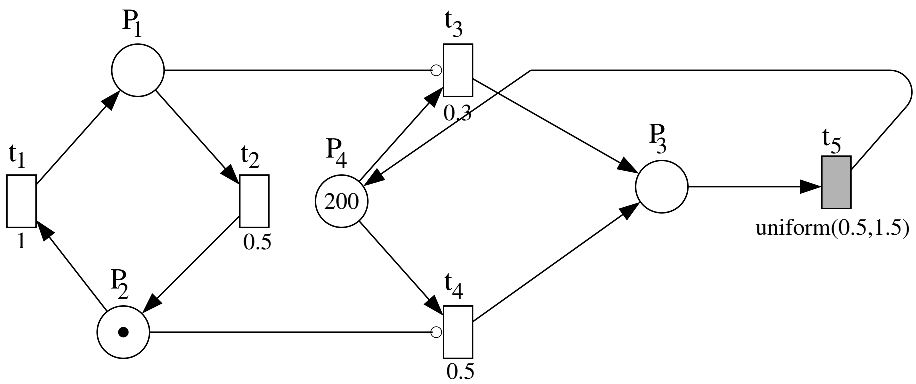

Figure 7.1: EDSPN model of a queueing system

The behaviour of the diffusion approximation shall be recapitulated in a Petri net model of a queueing system with Markov modulated arrival and uniform service time distribution. The discrete model has twice as many states as is the capacity of the queue (which correspond to the number of tokens in place P_4) and the fluid model has just two discrete states. In order to be able to solve the discrete model with TimeNET and the fluid model with the algorithms presented in chapter 6, in both the state space must be finite. As the utilization

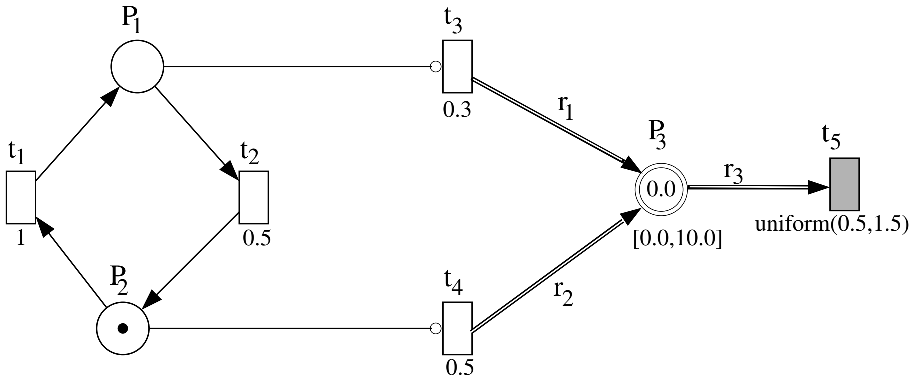

Figure 7.2: Second order FSPN model of a queueing system

approaches one the mean queue length gets large and the result obtained from the FSPN is expected to converge against the result computed with an EDSPN. The discrete model in its initial configuration is shown in figure 7.1. It is an EDSPN (extended deterministic and stochastic Petri net), for the service time is uniformly distributed. EDSPNs can not be mapped on a Markov process and no closed form solution exists. The results shown in figure 7.3 are computed with the stationary analysis component of TimeNET.

The approximation is carried out in replacing the discrete queue (place P_3) by a fluid one. This implies, that the arrival and service processes must be transformed into fluid flow rates as has been discussed in chapter 3. The fluid flow rates are normally distributed and the parameters of the normal distributions are computed as

$$\begin{aligned} \mu_{r_3} &= \frac{1}{E(u(a,b))} & \sigma^2_{r_3} &= \frac{2}{3}\frac{Var(u(a,b))}{(E(u(a,b)))^3} = \frac{2}{3}\frac{(b-a)^2}{(a+b)^3} \\ \mu_{r_1} &= \frac{1}{\lambda_3} & \sigma^2_{r_1} &= \frac{1}{\lambda_3} \end{aligned} \tag{7.1}$$

(see equation (4.3)), where λ_3 is the firing rate of t_3 and $u(a,b) = \text{Uniform}(0.5, 1.5)$. μ_{r_2} and $\sigma^2_{r_2}$ are defined analogously to μ_{r_1} and $\sigma^2_{r_1}$.

In order to vary ρ the mean service time is changed while its variance is kept constant. Hence, the series of uniform distributions is

$$\text{Uniform}(0.5 + x, 1.5 + x), \qquad x = 0.0, 0.1, 0.2, \ldots 1.2, 1.24 \tag{7.2}$$

Consequently, for the mean service time holds

$$\overline{x} = 1.0 + x \qquad x = 0.0, 0.1, 0.2, \ldots, 1.2, 1.24. \tag{7.3}$$

Note, that although the variance of the uniform distribution does not change while the expectation increases, the variance of the approximating normal distribution does change.

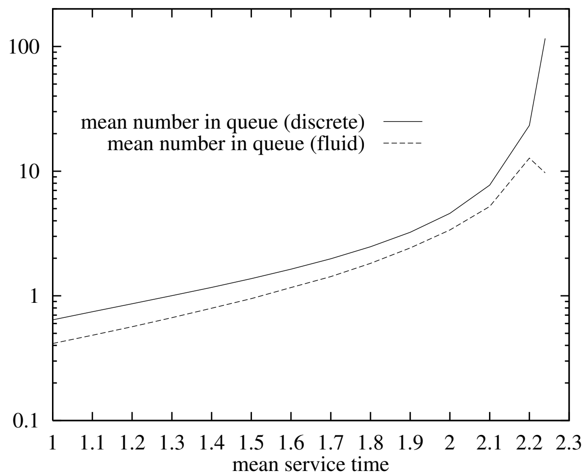

Figure 7.3: Mean number in the queue versus mean service time

In figure 7.3 the mean queue length in the EDSPN and the FSPN model are plotted versus the mean service time. The relative deviation of the two curves is plotted versus the

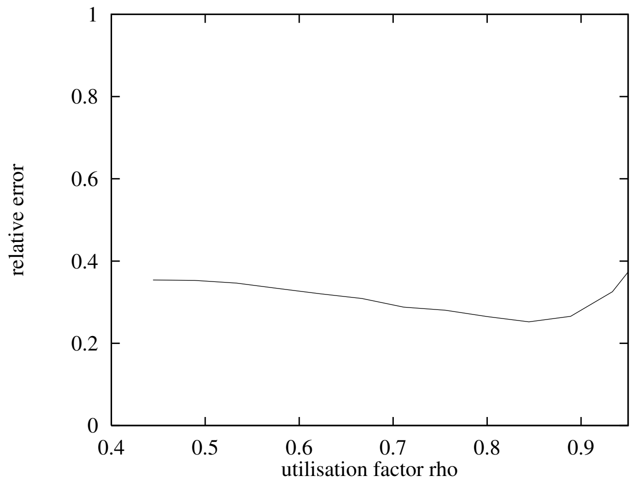

Figure 7.4: Relative error versus utilization

utilization factor ρ in figure 7.4. In steady state of the model for ρ holds

$$\begin{aligned} \rho &= \lambda \overline{x} \\ &= (P(\#P_1 = 1) \cdot \frac{1}{2} + P(\#P_2 = 1) \cdot \frac{1}{3}) \cdot \overline{x} \\ &= \frac{4}{9} \overline{x} \end{aligned} \tag{7.4}$$

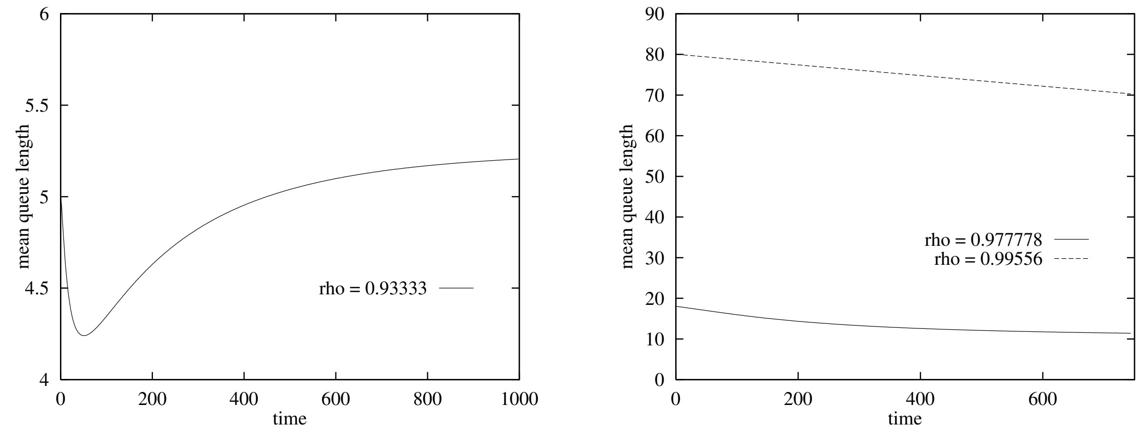

Figure 7.5: Mean length of the heavily loaded queue

Verifying the analytical results with the numerical solution scheme, however, is not easy. The difficulties are even larger since stationary results are needed but the stationary solution component is not available yet, so that the transient solution must be run for a sufficiently long time, so that the system is in steady state. For small values of ρ this is no

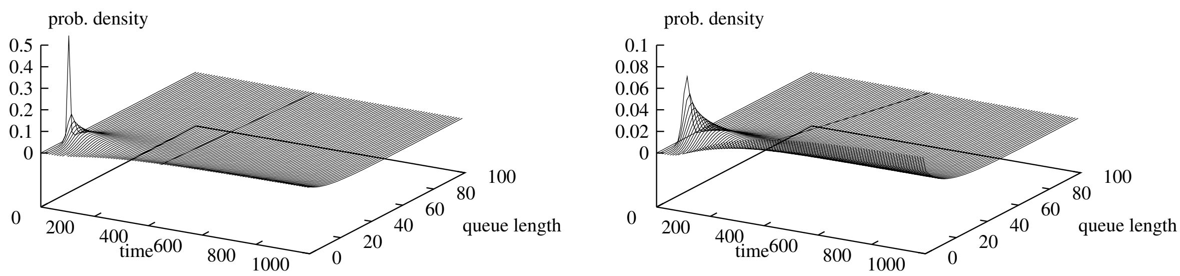

Figure 7.6: Probability density of the heavily loaded queue ($\rho = 0.9333$)

problem and the curve in figure 7.4 shows that if ρ approaches 0.85 from below the approximation becomes better. As ρ reaches 0.9 unfortunately the approximation quality again decreases. To explain that phenomenon the transient curves in figure 7.5 are included. They show, that even after 1000 time units no steady state is reached. For comparison reasons it should be mentioned that for $\rho = 0.444$ only 30 time units are needed to reach steady state. For $\rho = 0.977$ not even the slope downwards is completed. It becomes clear, that for $\rho > 0.9$ no usable mean values could be obtained.

Figure 7.6 shows the probability density for the fluid model and $\rho = 0.977778$. The density is extremely widespread, so that changes in the mean take place only very slowly.

7.2 A Performability Model

In performability modelling systems are analyzed whose components can fail and are repaired as discussed in chapter 1. The measures of interest are in these systems the reliability, that is the probability of the system being operational, and the performance, which is the amount of work carried out by the system in a certain time. Reliability is obtained from a Markov model in a straight forward manner, by summing the probabilities of the operational states in the model. Performance of the system is in the model the accumulated rate reward, which is harder to obtain. It is not a standard measure to be computed with a Petri net tool such as TimeNET [GKZH95]. In a Markov model the mean accumulated reward is computed as a weighted sum of the integral over equation (2.3) until the desired time point T. Fluid models are especially suitable for Markov reward modelling [WZ99].

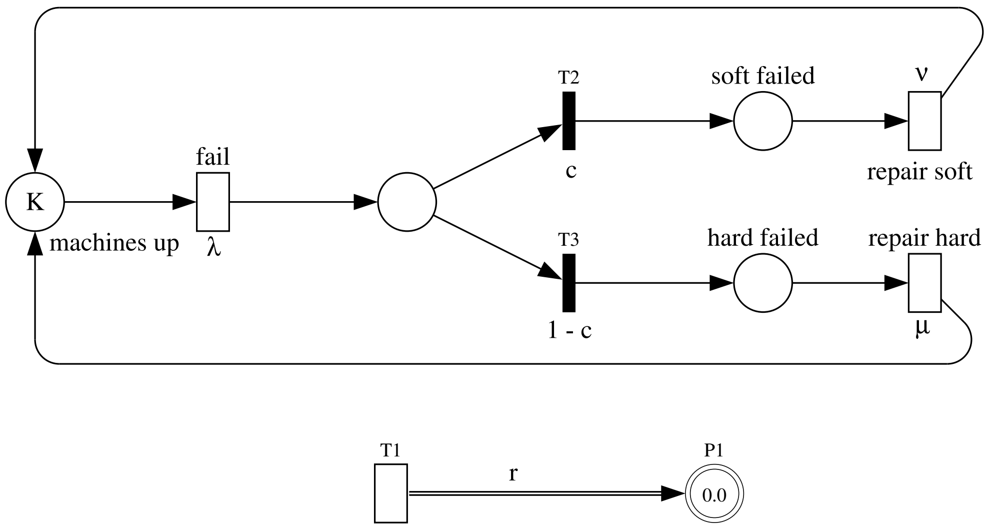

Figure 7.7: FSPN model of the example

The intention in presenting this example is to show, how easily accumulated reward can be modelled in a second order FSPN and not only its mean, but also its full density is obtained. The computation of the full density for accumulated reward has until now been an open problem in performability evaluation. In [Hor96] this application field has been identified, but no solutions have been presented.

The example that is regarded is a version of the extended machine-repairman (EMR) problem (see [vM94] p. 177). It is a typical example for a stiff Markov chain, since failure and repair rate may differ by orders of magnitude. This is, however, not the focus here, but none the less the oscillations of the solutions in figure 7.9 indicate a numerically difficult situation for early time points. Figure 7.7 shows a second order FSPN model of the EMR problem.

The data used here are the failure rates of the main generator in an aeroplane taken from [LS99]. So, it is assumed that there are K aeroplanes, the generators of which may fail independently. Each aeroplane provides a certain service and gains profit for its proprietor, the reward in the model.

Another application one can think of is an electricity producing reactor. Failure rates of components in reactors are still smaller than the failure rate of the generator used here. The reactor produces electricity and the owning company makes profit as long as the reactor is operational. In the FSPN model the fluid place holds the accumulated reward,

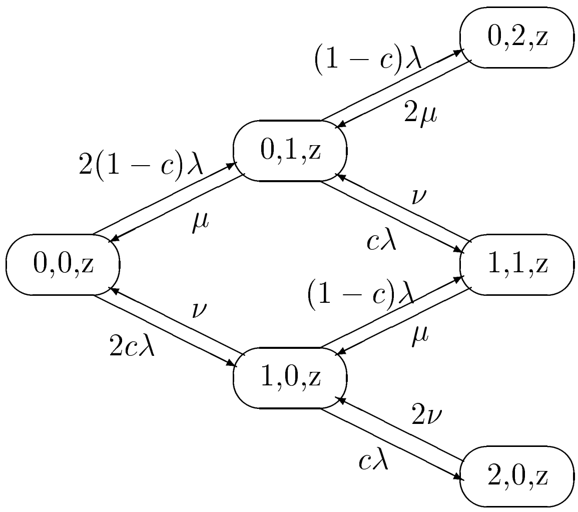

Figure 7.8: Reachability graph of the performability model

that is initially zero. For each operational machine a normally distributed amount of fluid (reward) with mean 1.5 and variance 0.5 is drained into the fluid place. There are K machines, each of which fails on the average after 670 operational hours ($\lambda = 0.03$ - measured in days of 20 hours each). Failure is a soft one with probability $c = 0.6$ and a hard failure with probability $1 - c = 0.4$. The repair of the soft failed machines takes on the average one hour ($\nu = 20$) and the repair of the hard failed machines takes on the average two hours ($\mu = 10$), both repair times being exponentially distributed. Note, that all timed transitions are *infinite server*, which means that the failure rate holds for each machine and there is a repair facility for each individual failed machine.

The reachability graph for $K = 2$ machines is shown in figure 7.8. The state space is defined as $\mathcal{M} = \{(i, j, z) | i = \#\text{soft failed}, j = \#\text{hard failed}\}$. The component z is included to stand for the continuous state space. The continuous state can take on many different values in each discrete state and is therefore represented by a variable.

To fully specify the FSPN the flow rate r must be defined. If all machines are failed, no reward is accumulated. In order that the differential equation is of parabolic type a small variance in all discrete states is required. The flow rate r is a normally distributed random variable with mean and variance as follows:

$$\mathbf{M} = \text{diag}(3.0, 1.5, 1.5, 0.0, 0.0, 0.0) \tag{7.5}$$

$$\mathbf{\Sigma}^2 = \text{diag}(1.0, 0.5, 0.5, 0.2, 0.2, 0.2) \tag{7.6}$$

Figure 7.9 shows the probability densities of the accumulated reward for all six discrete states. They sum up to the density over z. There is a clear drift and increasing variance visible in all plots. Note, that the plots are scaled differently and that the probability density for the first discrete state integrates to 0.99, which means that 99% of the probability mass remain in the first discrete state. A complete crash of the avionic system, which means entering the last three discrete states is highly unlikely, as also can be seen in the curve for the reliability of the system, shown in figure 7.10 on the right side.

The reliability of this system very quickly reaches an equilibrium state as is shown in the transient curve in figure 7.10. It never falls below 99.99% and therefore reward is accumulated at almost constant pace. The mean accumulated reward is computed in a discrete Markov model as

$$\begin{aligned}
\text{mean reward} &= \sum_{i=1}^{\#states} r_i \cdot \int_0^T \pi_i(t)\; dt \\
&= \sum_{i=1}^{\#states} r_i \cdot \int_0^T \pi_i(0) e^{Qt}\; dt \\
&\approx \sum_{i=1}^{\#states} r_i \cdot \left(\sum_{j=1}^{T/h-1} \pi_i(0) e^{Qjh} + \frac{\pi_i(0)}{2} + \frac{\pi_i(0)}{2} e^{QT} \right)
\end{aligned} \tag{7.7}$$

for some appropriate stepsize h. The last row is a simple numerical integration with the Trapezoidal rule (see [Sto94]).

The mean reward in a discrete model (7.7) has been computed with Mathematica for the performability model. Both the numerical solution of (7.7) and the mean accumulated reward obtained from the second order FSPN are shown in figure 7.10 on the left side.

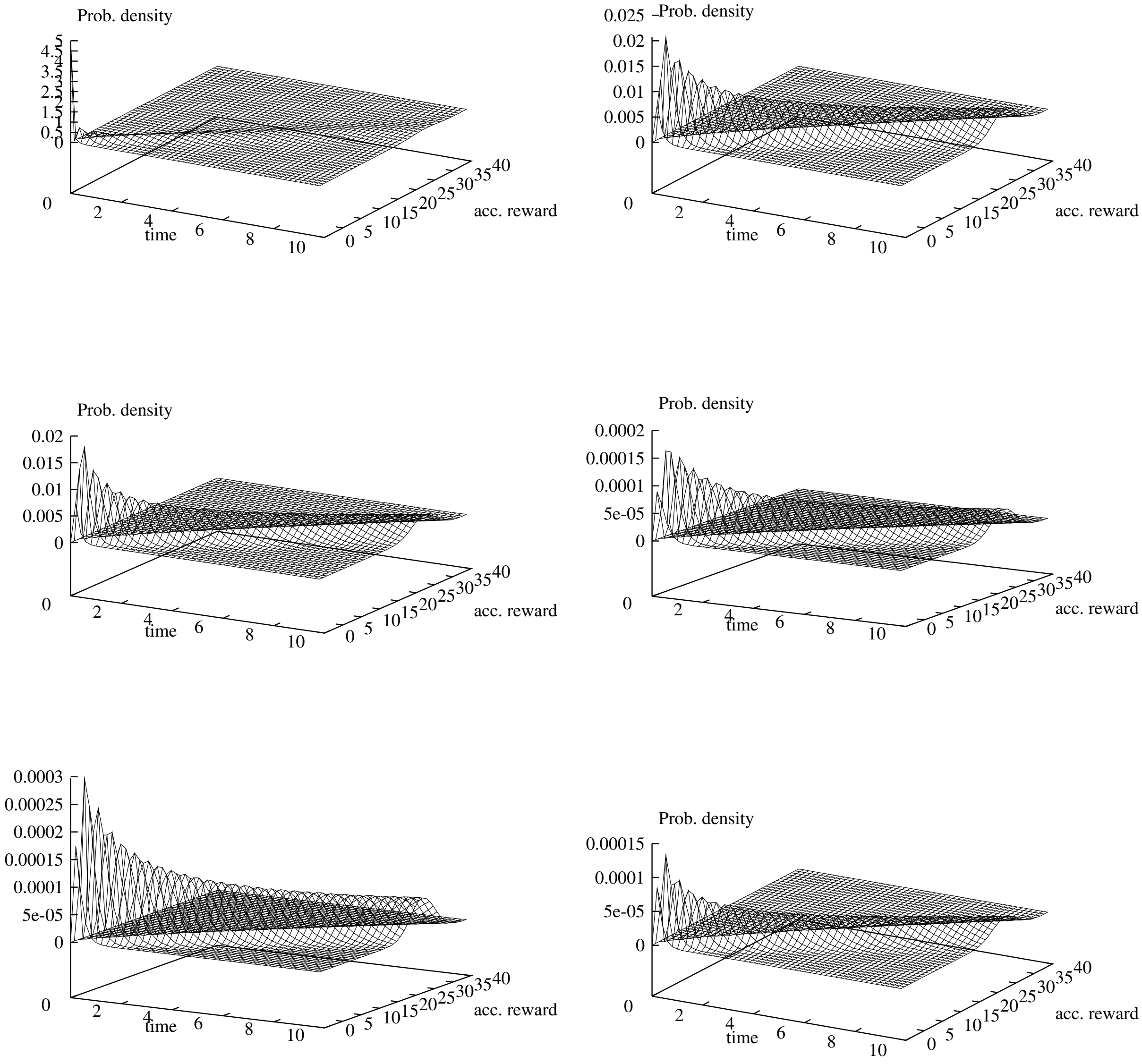

Figure 7.9: Probability densities for the discrete states

The two curves differ by absolute values of 0.17 to 0.2, depending on the time point. The relative error is at time zero 22% and decreases with time. After 10 time units it has gone down to 0.7%. This shows that with a second order FSPN accumulated reward can be modelled in an intuitive way and computed accurately along with the solution of the model. With the model time passing, the relative deviation between the fluid and the

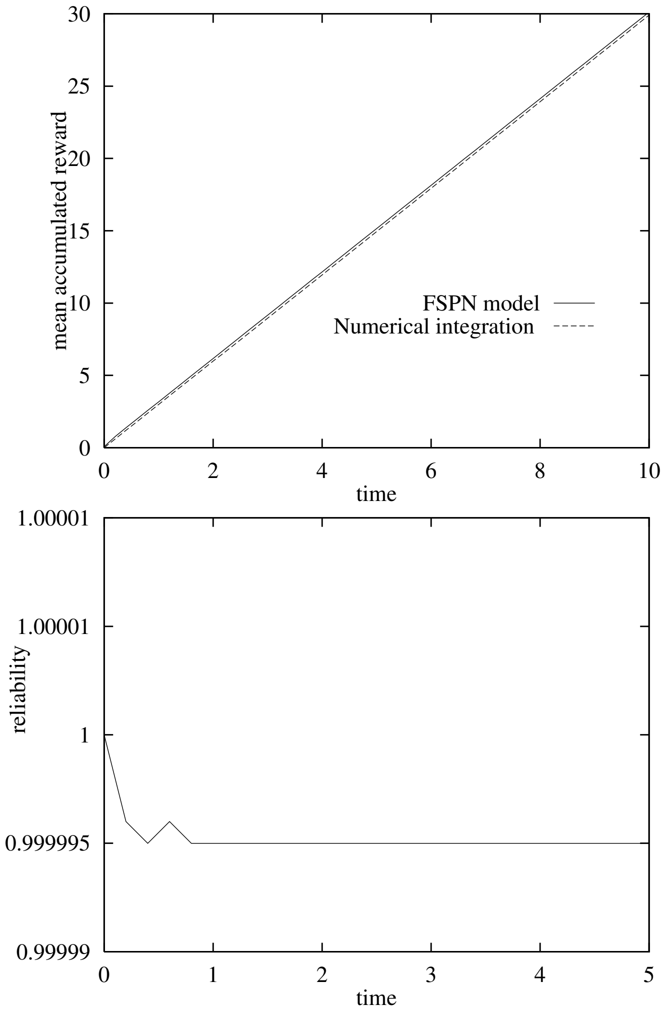

Figure 7.10: Mean accumulated reward and reliability of the system

discrete model decreases.

The little hook in the curve for the reliability at time 0.6 is most likely due to a round off error in the sixth digit.

7.3 A Hybrid System

In this section an intrinsic continuous system will be regarded, a controlled chemical batch reactor, taken from [SKPT98]. It consists in a liquid-cooled reactor with measuring, stirring and feed supply devices, and a controller. The controller is designed to realize the following sequence of steps: first, a certain amount of highly concentrated, dissolved substance B is fed into the reactor by opening valve v_B for a specified period of time. Then the stirrer motor M is switched on and a solution of substance A is added (by opening v_A) until the total reactor content has reached an upper limit. The chemical reaction starts, when v_A is opened. The exothermic reaction leads to an increase of the liquid temperature. To avoid dangerous heat up, the cooling is switched on when the temperature reaches some limit. The reactor is drained by opening valve v_o and the stirrer is switched off when the reaction time τ has elapsed. If the cooling is switched on, the inflow of substance A is stopped and

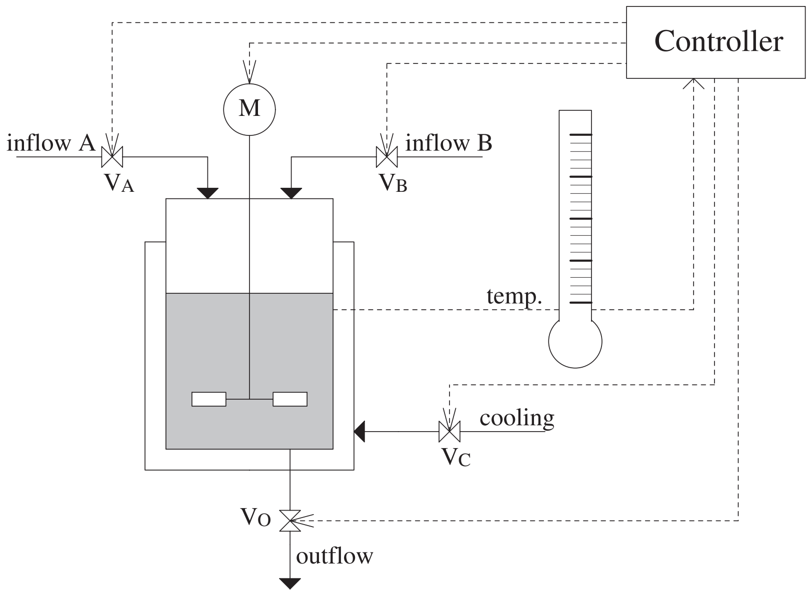

Figure 7.11: Sketch of the batch reactor

the reaction ends with a low concentration of substance A.

Only if the process ends with the desired concentration, the reactor is emptied and a new process can start. In figure 7.11 the system is sketched. The inflow of substance B takes on the average 30 minutes, the reaction time is on the average 25 minutes and the emptying of the reactor takes approximately 50 minutes. All durations are assumed to be exponentially distributed. The cooling is only switched on if the temperature exceeds some limit, L_1. If the temperature exceeds the second threshold L_2, with $L_2 > L_1$, the state of critical temperature is entered. So, depending on the continuous state a discrete state transition of the controller takes place.

The second order FSPN model is shown in figure 7.12. The transitions are labelled with the original rates of the exponential distribution, which will be scaled by factor 10 for the numerical solution.

The reachability graph is depicted in figure 7.13. It has six discrete states, the states of the controller, two of which are absorbing states. The states are labelled with the names of the discrete states, which are the states of the controller. Added to each discrete state are two continuous dimensions, the reactor content and the temperature. They are omitted in the reachability graph. The numbers in the small circles are used for referencing the solutions in the single states.

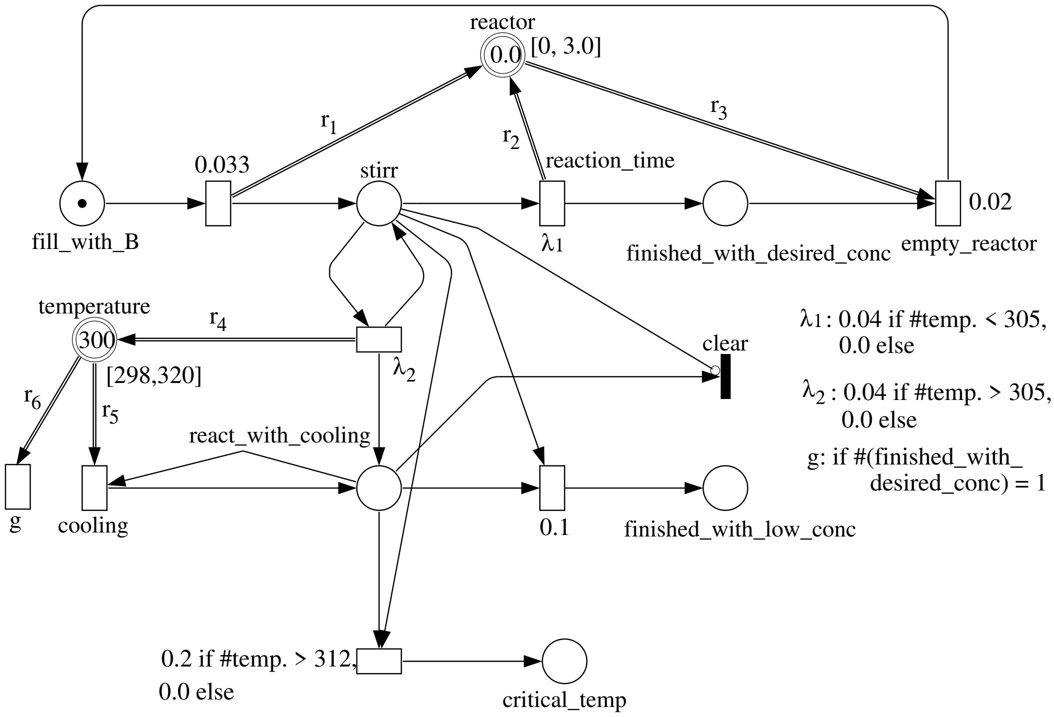

Figure 7.12: FSPN model of the batch reactor

The fluid flow rates are defined through expectation and variance. For the reactor in- and outflow the rates are measured in cubic meter and are defined as deterministic rates ($\sigma_i^2 = 0$).

$$\mu_{r_1} = 0.05 \qquad \mu_{r_2} = 0.04 \qquad \mu_{r_3} = 0.06 \tag{7.8}$$

The temperature is measured in Kelvin and the rates, at which the temperature changes are

$$\begin{aligned} \mu_{r_4} &= 0.08 \cdot (\#reactor)^2 & \sigma_{r_4}^2 &= 0.1 \\ \mu_{r_5} &= 0.04 \cdot (\#reactor)^2 & \sigma_{r_5}^2 &= 0.1 \\ \mu_{r_6} &= -0.05 & \sigma_{r_6}^2 &= 0.05. \end{aligned} \tag{7.9}$$

All the parameters in the model have been scaled by factor 10 for numerical convenience and it follows for the parameter matrices

$$
\begin{aligned}
\mathbf{M}_1 &= \mathrm{diag}(0.5,\ 0.4,\ 0.0,\ -0.6,\ 0.0,\ 0.0) \qquad \mathbf{\Sigma}_1^2 = \mathbf{0} \\
\mathbf{M}_2 &= \mathrm{diag}(0.0,\ 0.8z_1^2,\ 0.4z_1^2,\ -0.5,\ 0.0,\ 0.0) \\
\mathbf{\Sigma}_2^2 &= \mathrm{diag}(0.0,\ 1.0,\ 1.0,\ 0.5,\ 0.0,\ 0.0)
\end{aligned}
\tag{7.10}
$$

The generator matrix is scaled analogously. The model has been solved over 20 time units, which corresponds to 200 minutes in the original model. The fully implicit scheme ($\theta = 1$) has been used and the stepsizes have been set to $h_1 = 0.1$, $h_2 = 0.1$ in space and $k = 0.001$ in time. The stepsizes in space are rather large for the complexity of the problem, so at early times the solution might not yet be very smooth.

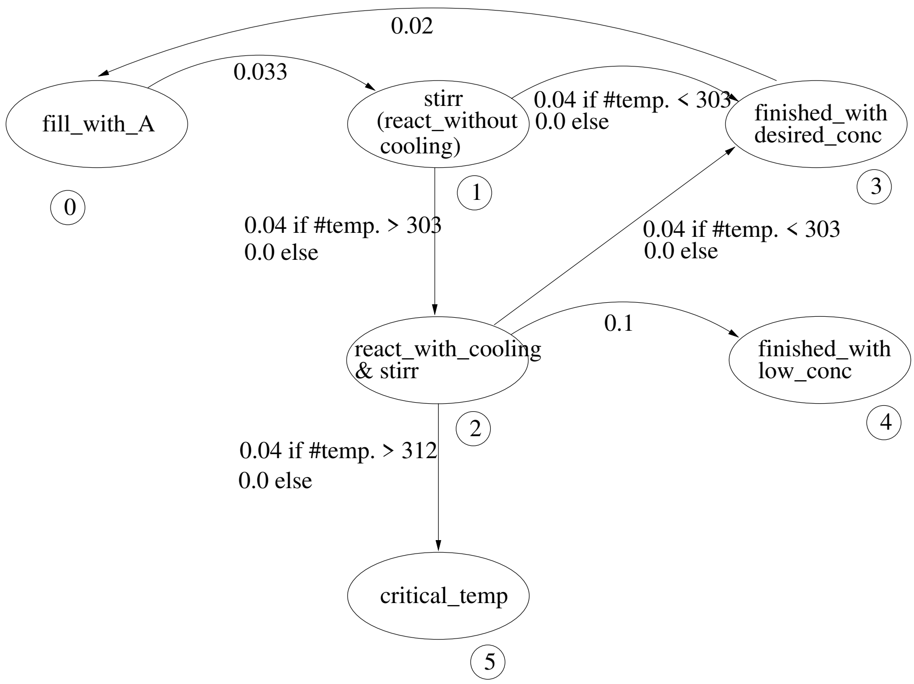

Figure 7.13: Reachability graph of the FSPN model

The solution of the differential equation gives a two-dimensional density, that changes over time. It cannot be represented in one plot, only in a series of plots for different time points, or in a film. In the following the transient evolution of the densities[1] in all discrete states are shown at selected time points. The chosen instants are t = 0.5, 1.0, 4.0, 8.0, 12.0 and 20.0, which corresponds to the system behaviour after 5, 10, 40, 80, 120 and 200 minutes. In the states two, four and five there is probability mass only after at least 4 time units. Therefore the first two plots are omitted for these states. The labeling in the plots refers to the time units of the solution procedure.

[1] Strictly speaking this is a density only after normalization. Nonetheless, we will call the fractions of a density also density.

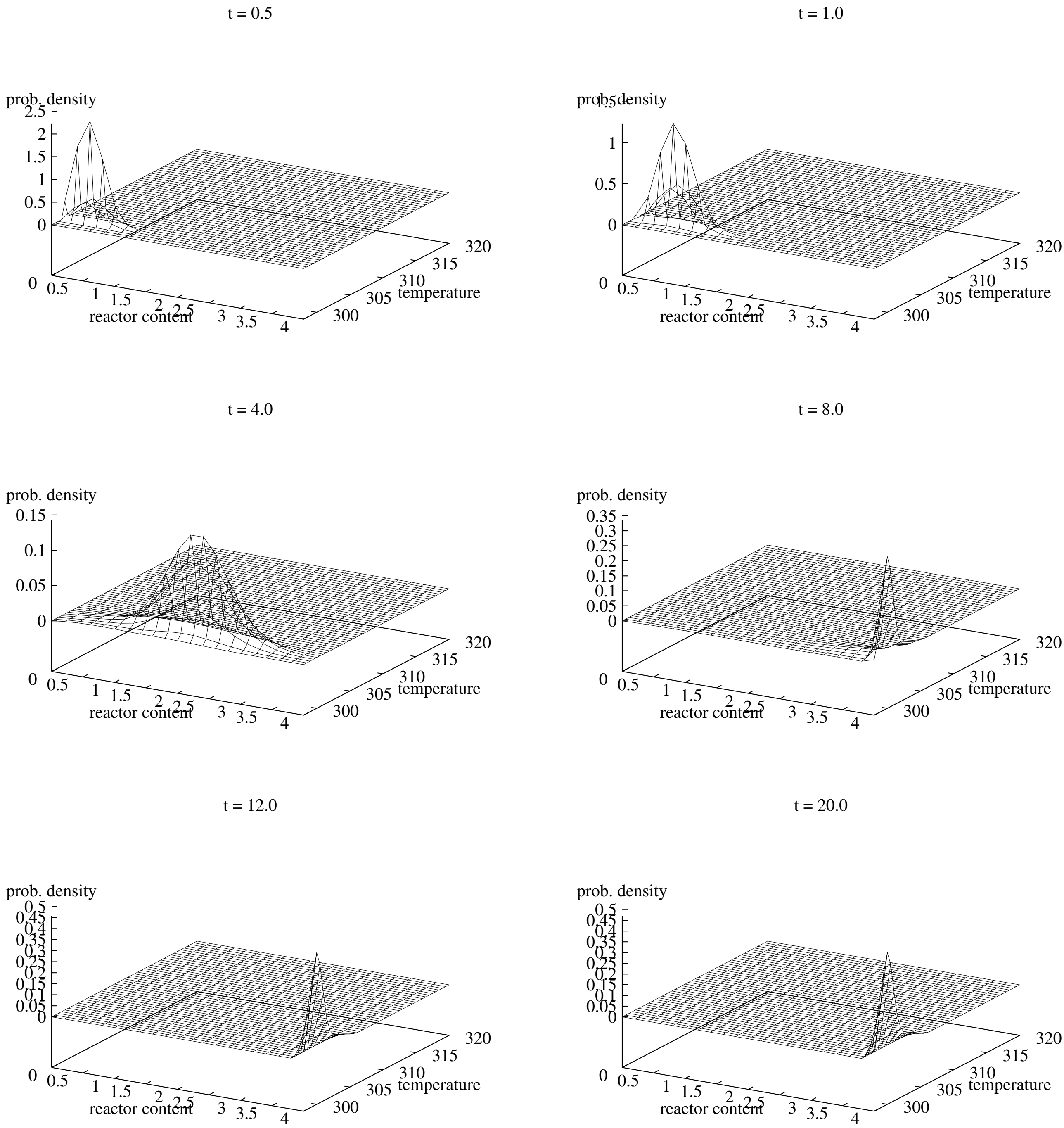

Figure 7.14: Probability density at selected time points in state *fill_with_B* (state 0)

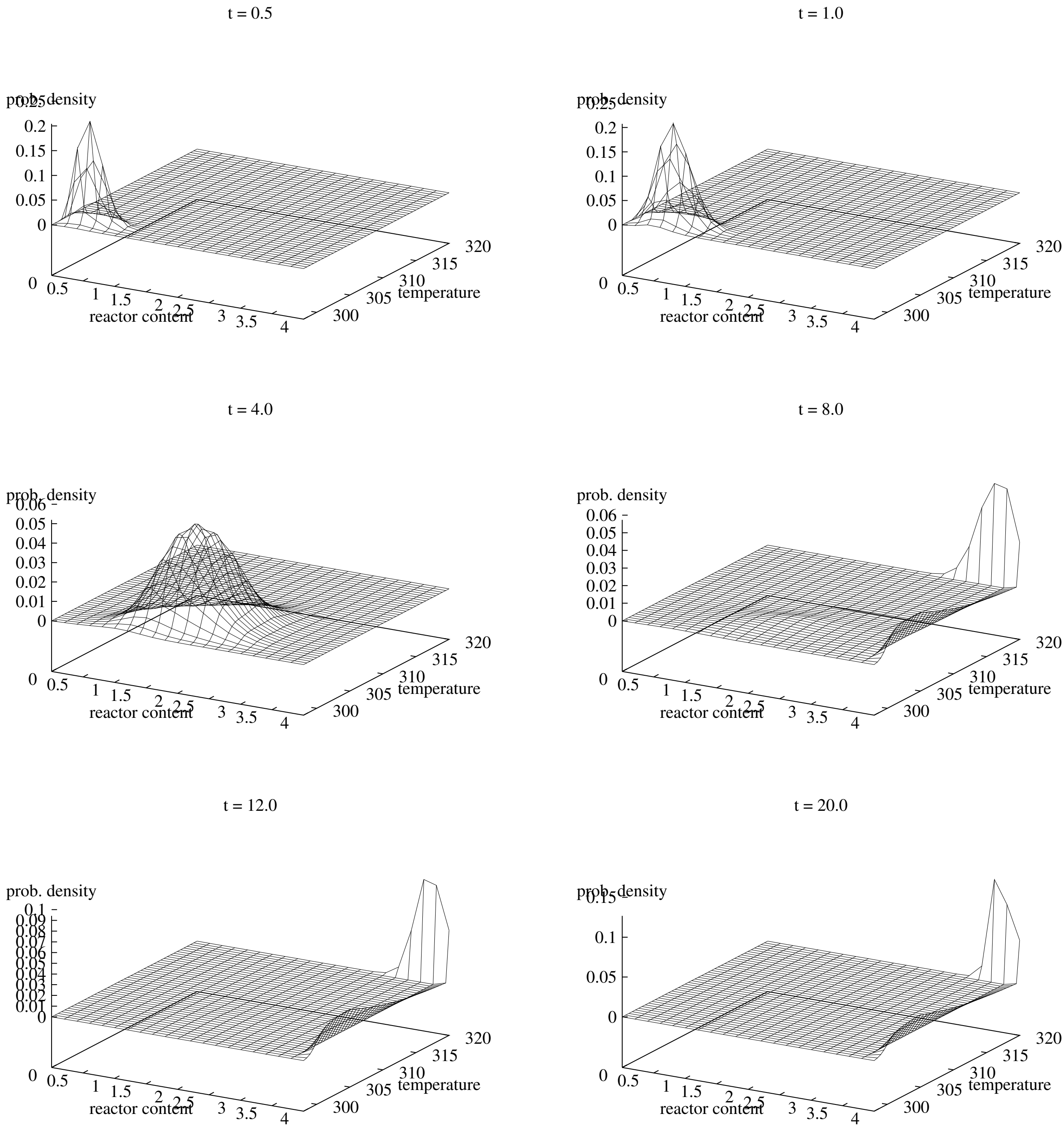

Figure 7.15: Probability density at selected time points in state *react_without_cooling and stirr* (state 1)

In figure 7.14 on page 118 the evolution of the density in the discrete state zero (*fill_with_B*) is shown. In state zero the reactor is filled with substance B. When the system is in this state the temperature always remains below the first threshold. The reactor fills up quickly and soon is full with high probability, whenever the controller is in state zero. In state one, the plots for which are shown in figure 7.15 on the preceding page substance A is added. In this state too, after the initial phase of 120 minutes (12 time units) the reactor is full with high probability. The presence of both substances leads to a chemical reaction and this causes increasing temperature, which starts roughly after 4 time units.

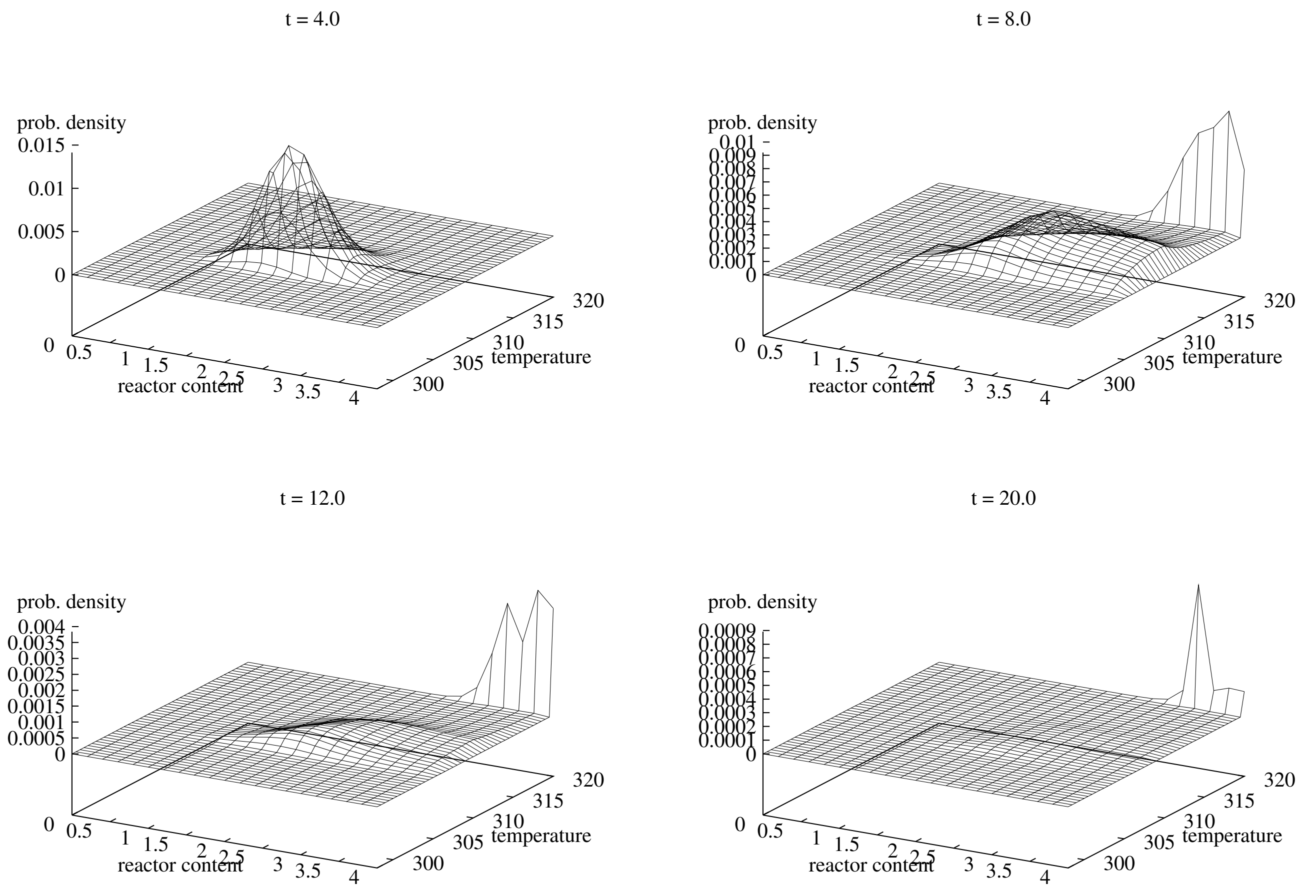

Figure 7.16: Probability density at selected time points in state *react_with_cooling* (state 2)

Two observations can be made for the system's evolution in state 2 (*react_with_cooling*), which is shown in figure 7.16: this state is entered only, if the temperature is at least 305K (above L_1) and it is a transient state. After 20 time units the probability density in this

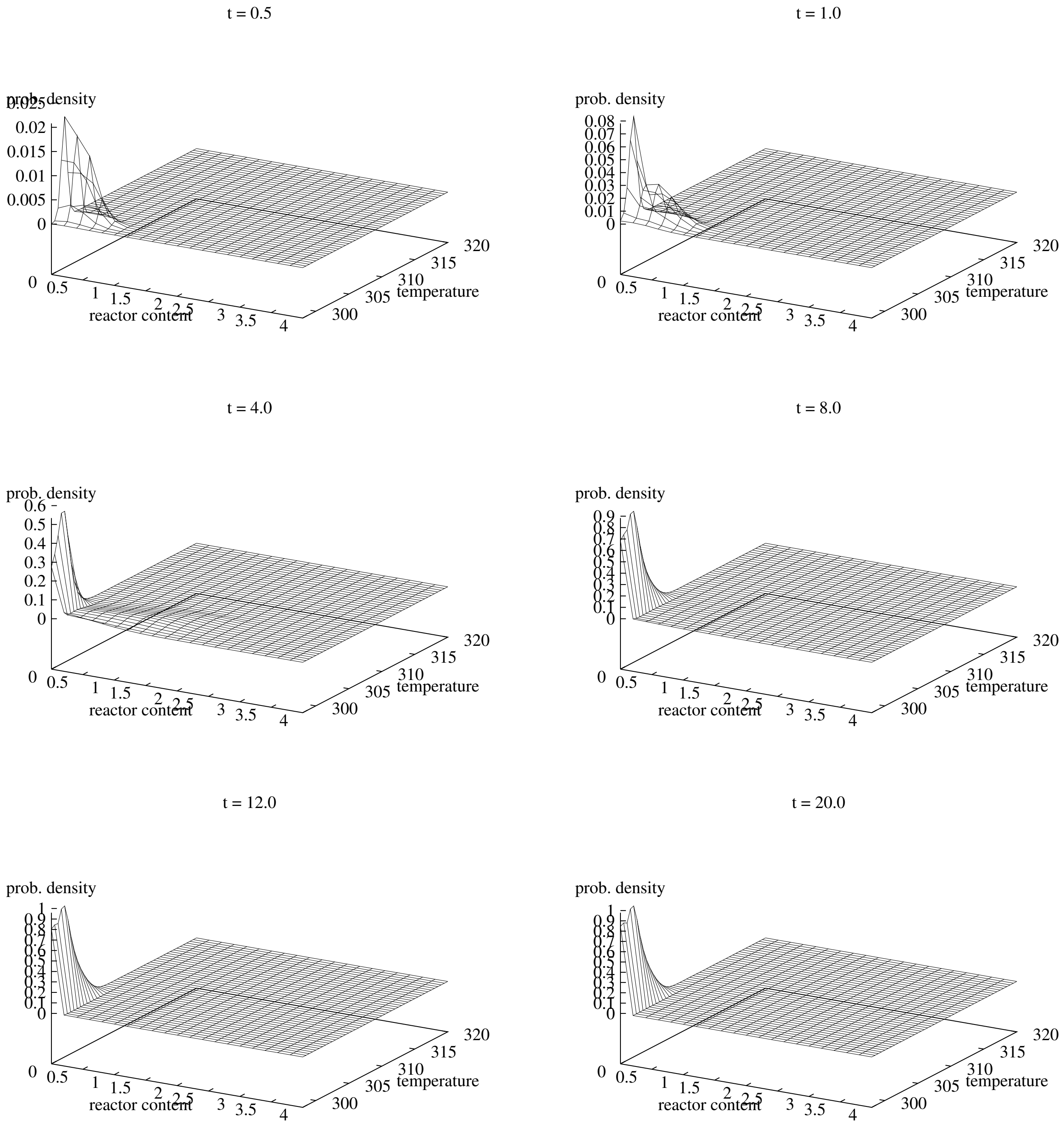

Figure 7.17: Probability density at selected time points in state *finished_with_desired_conc* (3)

state integrates to only 0.00032 and it is constantly getting less, while after 5.5 the probability mass in this state was equal to 0.1. This effect can also be seen in figure 7.22 on page 126.

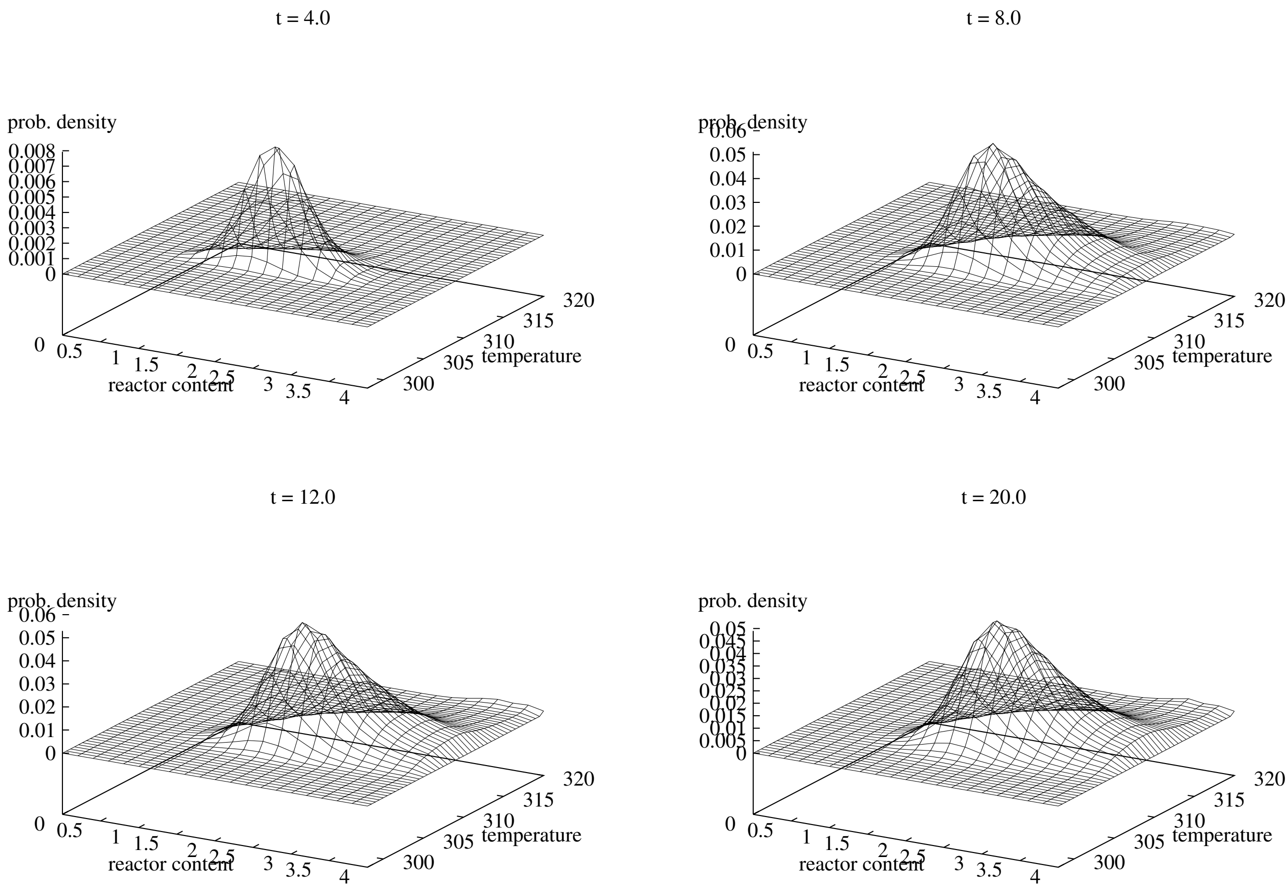

Figure 7.18: Probability density at selected time points in state *finished_with_low_conc* (4)

In state 3 (*finished_with_desired_conc*) the desired solution is obtained, the temperature remains below the lower threshold and the reactor is drained. When the system is in state 3 the reactor is empty with high probability as can be seen in figure 7.17 on the page before.

State 4 (*finished_with_low_conc*) is entered only if the temperature increases beyond the first threshold ($L_1 = 305$). In this state the reactor is not emptied again, which can also be seen in the plots in figure 7.18.

State 5 (*critical_temp*) is only reached when the second threshold ($L_2 = 312$) is crossed.

The plots are shown in figure 7.19. State 5 is entered later in time than state 4 and in state 5 the reactor contains more liquid than in state 4.

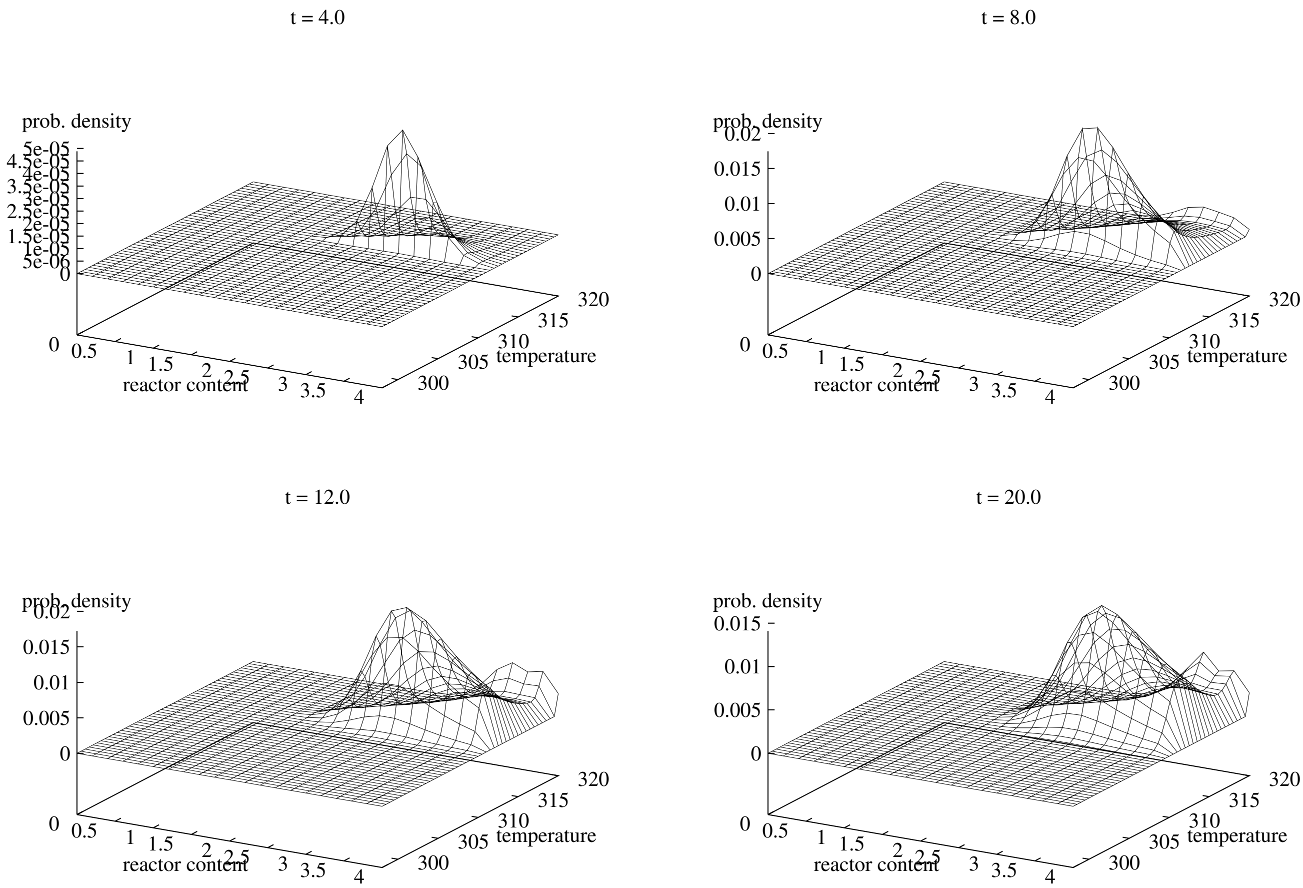

Figure 7.19: Probability density at selected time points in state *critical_temp_* (5)

Summation over all discrete states gives the density for the overall system, as shown in figure 7.20 on the following page. It gives insight in the evolution of the continuous variables, no matter in what state the controller is. The behaviour of the system is at first dominated by state 0, the initial state, and then also by states 3 and 4.

Integrating over one continuous dimension gives the marginal density with respect to the other continuous dimension. Figure 7.21 on page 125 shows all transient curves for the marginal densities in the different discrete states with respect to the temperature. Here again it can be seen (as in figure 7.22 on page 126) that state 2 is very unlikely to be reached again after 10 time units and states 2 to 5 are only reached when the temperature stays in some range.

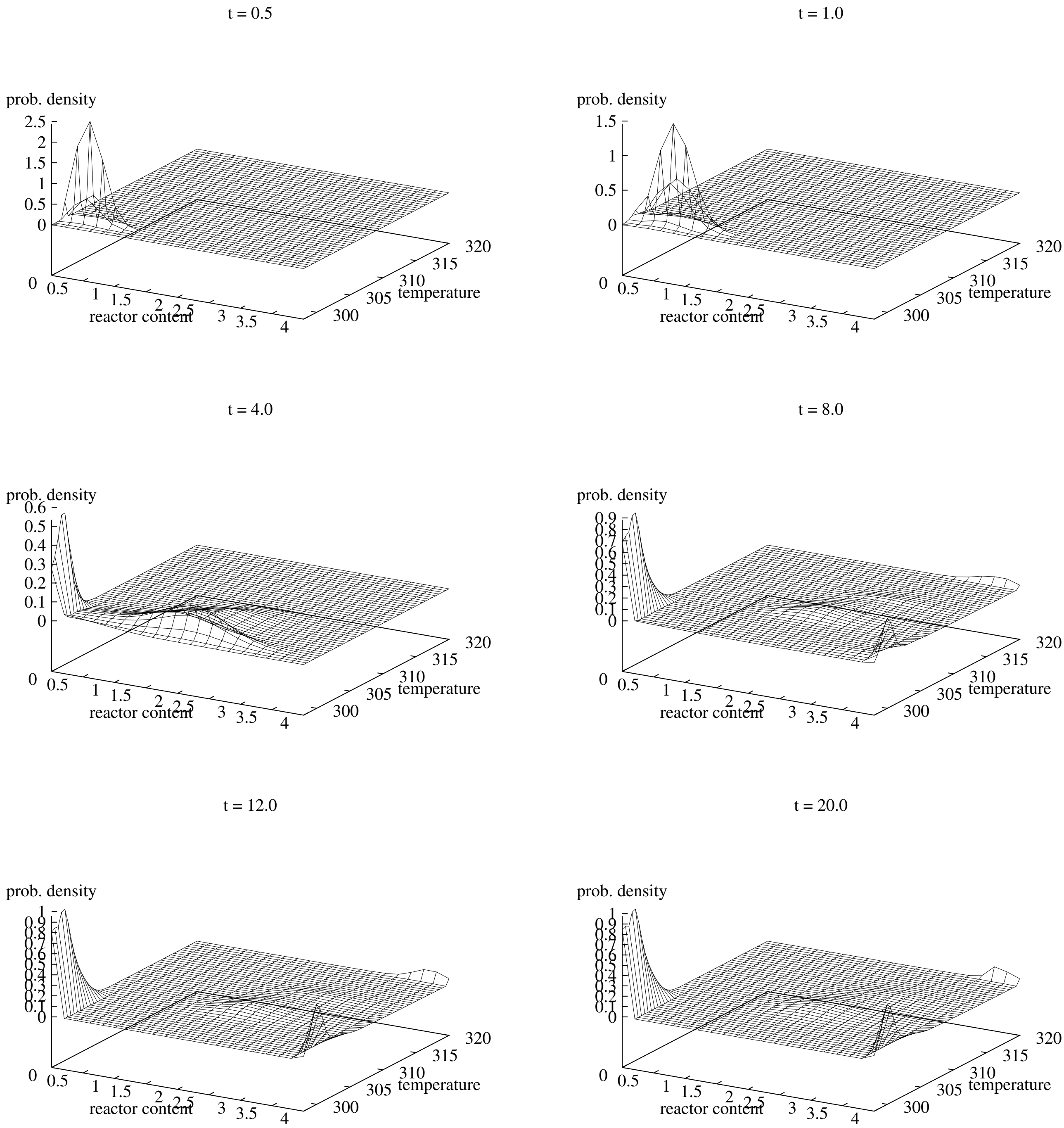

Figure 7.20: Probability density at selected time points for the whole system

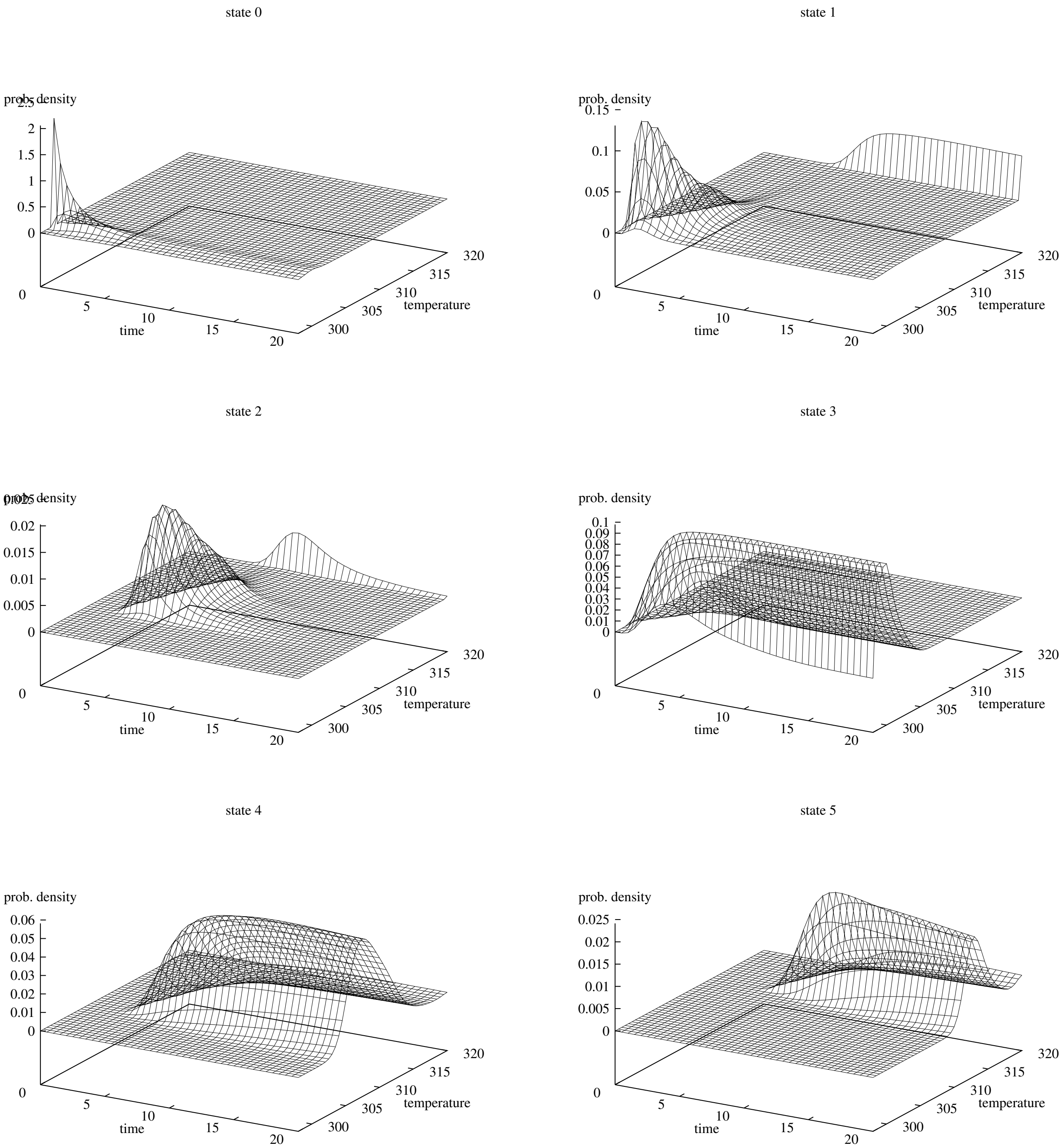

Figure 7.21: Transient marginal probability density

Integration over both continuous dimensions gives the discrete state probability. It is shown for all discrete states in figure 7.22 in the right plot. After 20 time units the probability of dangerous heat up is 8.8%. The probability that the cooling is needed is 53% (it is computed as the sum of the state probabilities of states zero, one and three).

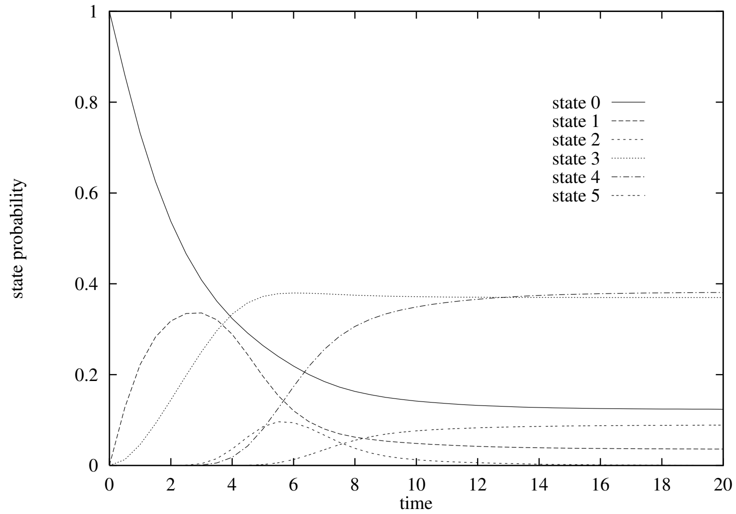

Figure 7.22: Probability density and state probabilities

The transient evolution of the mean in either continuous direction is shown in figure 7.23. The mean reactor content straight away increases until it reaches its equilibrium after approximately 10 time units. The mean temperature only begins to increase considerably after 2-3 time units, which corresponds to 20-30 minutes in the system. This is the time needed to fill in substance B.

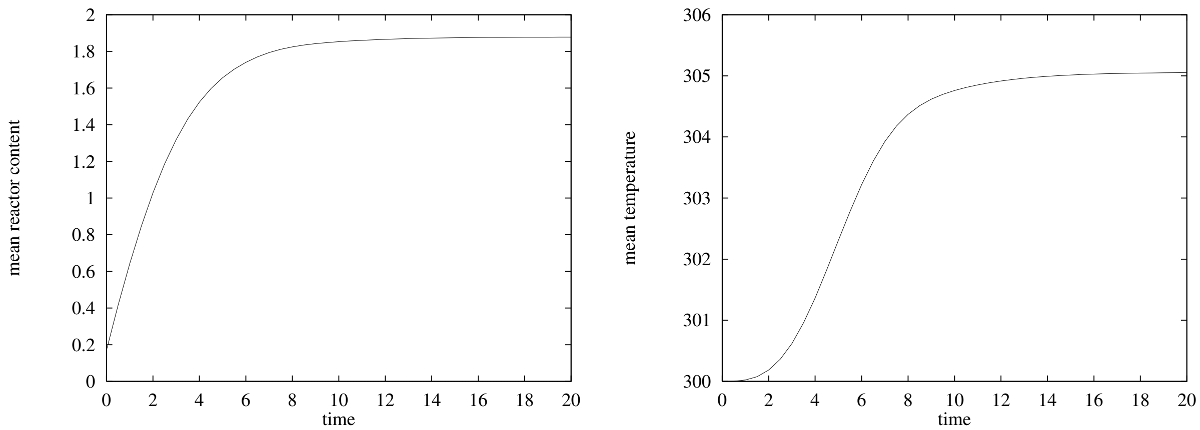

Figure 7.23: Mean reactor content and mean temperature

7.4 Unfinished Work in a Queueing System

Unfinished work in different kinds of queueing systems can be modelled with an FSPN including fluid jump transitions [Wol99]. On the other hand in some cases there also exists an analytical formula for the mean unfinished work. In order to study the behaviour of the jumps both the FSPN and the analytical queueing theory results will be compared in this section.

For the following considerations unfinished work denotes the amount of work that must be performed in order to empty the queue. Hence, the customer in service is not taken into account. Unfinished work in a different queueing system has already been regarded in [Wol98b].

First a queueing system with one arrival stream and one service unit will be regarded. Then the arrival process will be replaced by a more complex one, where two classes of customers arrive in an MMPP. In this example the different boundary treatments defined in chapter 4 will be studied. In order to be able to obtain results with TimeNET and the algorithms presented here the parameters will be chosen within the same order of magnitude.

An MMPP as an arrival process is often used when modelling Internet traffic, which is characterized by rates that differ enormously. In order to demonstrate the capability of the algorithm, finally in this section a realistic set of parameters will be used.

The system considered first has just one arrival stream and one service unit. The service times are at first exponentially distributed then uniformly distributed and finally deterministic service times are used. The capacity of the fluid place has been chosen sufficiently large so that the system is equivalent to an unrestricted one. The z-axis has length 40, that is $z^{\min} = 0, z^{\max} = 40$. The discrete and the fluid model for the M/G/1 system are shown in figure 7.24.

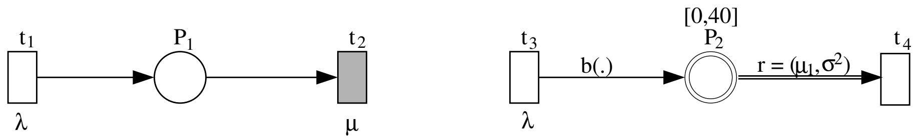

Figure 7.24: Discrete and fluid Petri net model

The arrival rate, that is the firing rate of the transitions t_1 and t_3, is set to $\lambda = \frac{1}{2}$. The parameters of the service time distribution are chosen in such a way that the utilization ρ of the system remains constant for all service time distributions. At first, the model with the exponentially distributed service time is regarded. In the discrete model the service rate, which is the firing rate of t_2 is set to $\mu = \frac{2}{3}$ (and consequently $\rho = \frac{3}{4}$). The corresponding fluid model has an exponential jump height distribution with rate μ as well. So, $b(.) \cong \text{Exponential}(\mu)$. The number of tokens in P_1 is the number of customers in the system, whereas the amount of fluid in P_2 corresponds to the amount of unfinished work in the system (or the virtual waiting time).

The flow rate r along the fluid arc, that is specified by expectation and variance $(1, \sigma^2)$, should be deterministic to construct an equivalent fluid model to the discrete one. To show this effect the variance is reduced stepwise in subsequent runs and it is expected, that the results come closer to the analytical ones the smaller the variance becomes.

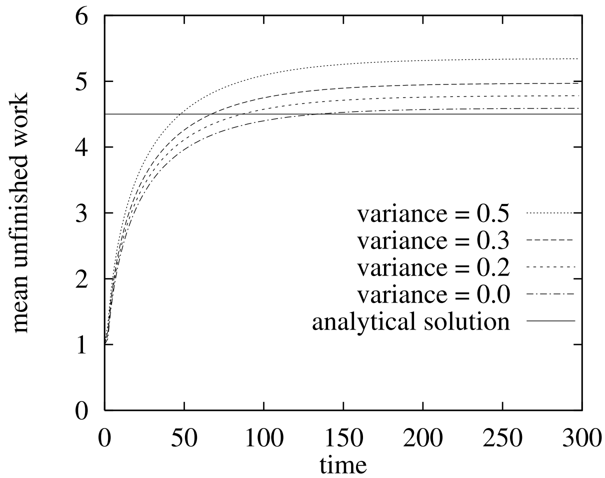

Figure 7.25: Mean unfinished work in an M/M/1

The analytical results are only stationary expected values, so the mean of the content of the fluid place is computed over time and this measure is expected to converge against the analytical result with time. The mean unfinished work in an M/M/1 queue is

$$\overline{U} = \frac{\rho/\mu}{1-\rho} = 4.5, \tag{7.11}$$

for the values of the parameters chosen here. Furthermore the following equalities hold

$$\overline{U} = \frac{\overline{N_q}}{\lambda} = \frac{\overline{N} - \rho}{\lambda} = \frac{\overline{N}}{\mu}. \tag{7.12}$$

In (7.12) the first equation from the left is Little's law. The third term explains the mean number in the queue as the mean number in the system minus ρ customers in service, on the average. This is, surprisingly, equal to the mean number in the system times the mean service time, which could also be understood as the amount of unfinished work in the system.

Figure 7.25 shows the mean unfinished work for the M/M/1 fluid model and with reducing the variance of the flow rate the stationary mean fluid level seems to approach nicely the analytical solution.

In order to receive the same utilization as for the M/M/1 queue the parameters for the uniform(a,b) distribution have been chosen $a = 1.0, b = 2.0$. The analytical result for the unfinished work in an M/G/1 queue is given by the Pollaczek-Khinchin formula and evaluates to

$$\overline{U} = \frac{\lambda \overline{x^2}}{2(1-\rho)} = 3, \tag{7.13}$$

for the parameters chosen here.

The mean unfinished work for uniformly distributed service is shown in figure 7.26. For very small variance the analytical solution is here underestimated. An explanation for this phenomenon could be, that the approximation quality of the fluid model depends on the squared coefficient of variation of the service time distribution. It has been found out in [RK74] that values of the scv less than one lead to an underestimation of the true results and that the exponential distribution with scv = 1 is one of the best to approximate. The uniform $(1.0, 2.0)$ distribution has scv $= \frac{1}{27}$.

At last, deterministic service time with a durations of 3/2 time units has been used. The mean stationary amount of work in the system is

$$\overline{U} = 2.25.$$

The deterministic service time has no variance and herewith the lowest result for $\overline{U}$. The results from the fluid model are shown in figure 7.27. Again, the results from the fluid model approximate the analytical ones from above with decreasing variance of the fluid flow out of the fluid place.

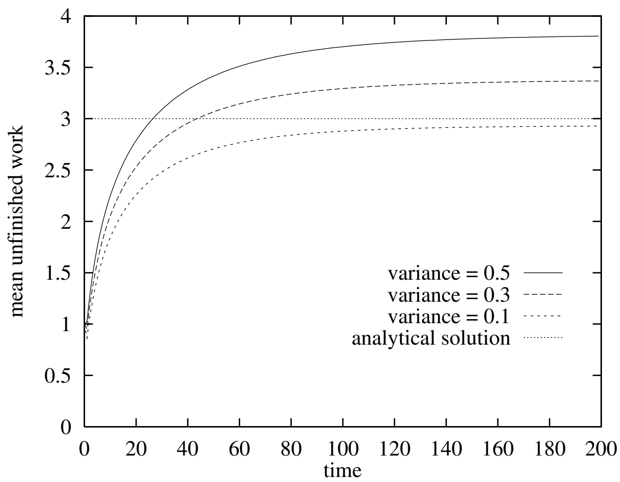

Figure 7.26: Mean unfinished work in an M/G/1 system

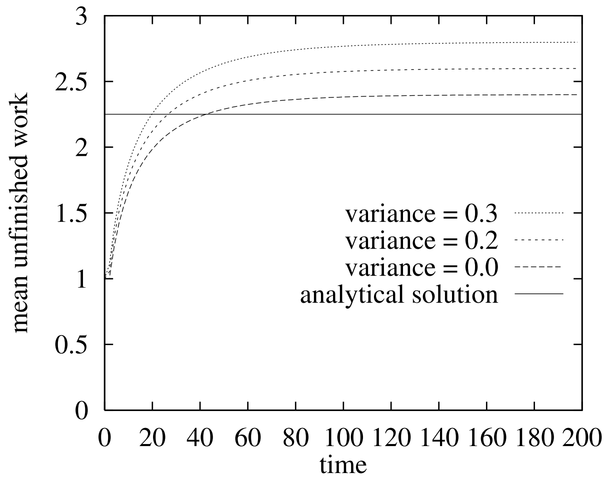

Figure 7.27: Mean unfinished work in an M/D/1 system

For this example finally the full probability density is shown for the exponential service time distribution and the variance of the outflow equals 0.2. All the densities look similar to this one, therefore only one plot is included. The densities differ in the probability of having an empty fluid place which is less if the variance increases. Also the slope in the densities over the fluid level becomes softer with larger variance.

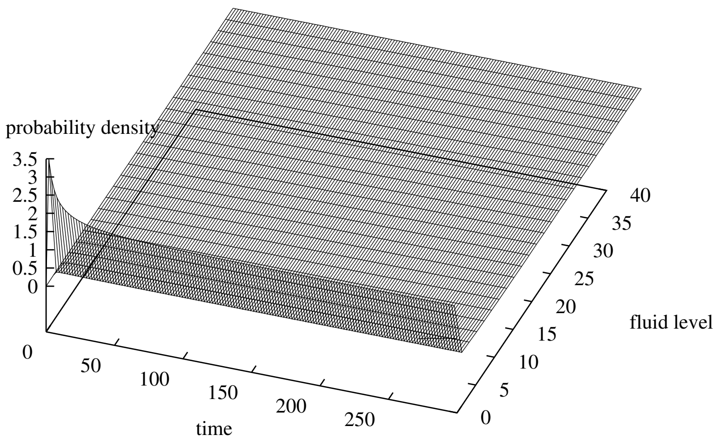

Figure 7.28: Full density of the unfinished work

Another model shall be regarded in order to study the two different boundary treatments for the jumps. Again a queueing system is modelled, but the arrival process is modelled with an MMPP and the capacity of the fluid place is reduced to 10, so that the upper bound is reached by the fluid level. The interpretation of the treatments at the boundary is, that the system can hold only a certain amount of work. If the system is almost full an arriving customer can either be rejected, or provided with a limited service.

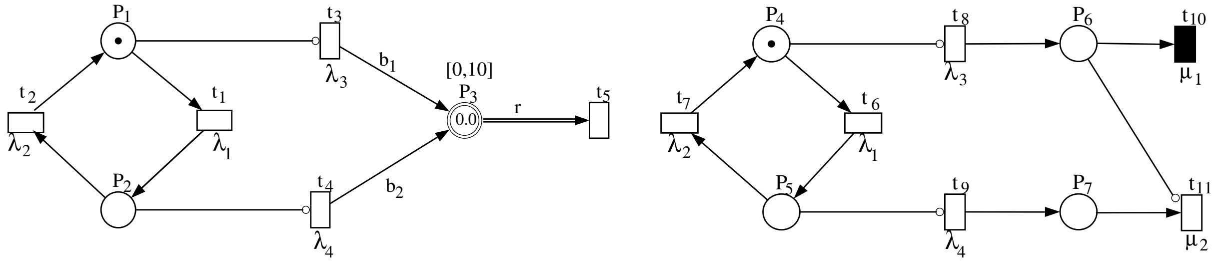

Figure 7.29: DSPN and FSPN model of a queueing system with two classes of customers

Again the amount of unfinished work is computed. Figure 7.29 shows on the left the second order FSPN and on the right the corresponding DSPN. In this model different types of customers are processed, those with exponentially distributed service time and those with uniform service time. A model where in addition priority classes have been taken into account has been studied in [Hal72, HS72]. There it was possible to find a mapping onto a M/G/1 queue and solve for the queue length distribution using Laplace transforms.

In order to simulate a single server queue in the DSPN the deterministic transition is given priority and only one of the two server transitions t_{10} and t_{11} is enabled at a time.

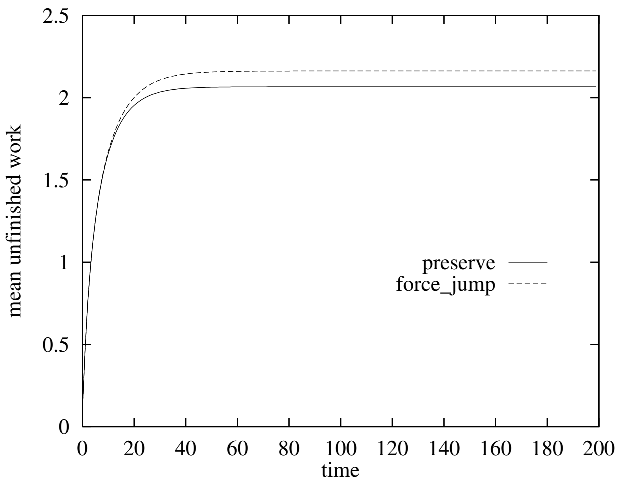

Figure 7.30: Mean unfinished work for the two boundary treatments

The rates of the modulating process are $\lambda_1 = 0.5$ and $\lambda_2 = 0.4$. If a token is in P_1 customers arrive at rate $\lambda_4 = 0.6$ and their service time is deterministic of length $b_2(.) = Dirac(1.5)$.

If on the other hand the token is in P_2 the customers arrive at rate $\lambda_3 = 0.5$ and they have an exponentially distributed service time with rate $b_1(.) = \text{Exponential}(1.0)$. Transition t_5 is there only to enable the outflow along the fluid arc, that has expectation 1.0 and variance 0.2. Again the fluid place holds the amount of unfinished work in the system, that is now composed of different types of work, the deterministic and the exponentially distributed.

In the discrete model the modulating process is the same as in the fluid one, but the queue must be split into two (represented by the places P_6 and P_7) because there is no other way to handle different types of customers. Transition t_{10} represents the exponential service time and transition t_{11} the deterministic service. So for the calculation of waiting time, or number of customers in the system the content of the places P_6 and P_7 must be added to gain the overall queue length.

The mean amount of unfinished work is shown in figure 7.30 for both boundary strategies. It is again computed as the mean content of the fluid place. As expected the strategy preserve keeps the fluid level lower than the strategy force jump.

There is in general no analytical formula for the mean unfinished work in this kind of a queueing system. An upper limit is, however, given by the average

$$av = \frac{\lambda_2}{\lambda_1+\lambda_2}\frac{\lambda_4\overline{x^2_{b_2}}}{2(1-\lambda_4\overline{x}_{b_2})} + \frac{\lambda_1}{\lambda_1+\lambda_2}\frac{\lambda_3\overline{x^2_{b_1}}}{2(1-\lambda_3\overline{x}_{b_1})} = 3.71, \tag{7.14}$$

and $\overline{U} < av$. Equation (7.14 is a weighted sum over the earlier introduced Pollaczek-Khinchin formula.

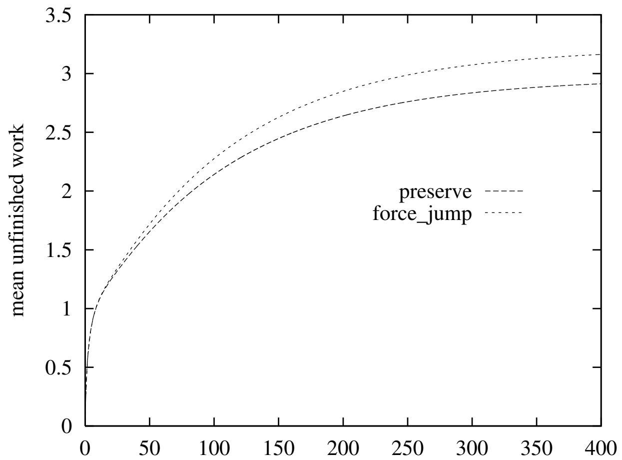

Figure 7.31: Mean unfinished work for the two boundary treatments (scaled parameters)

If the rates λ_1 and λ_2 converge to zero and their ratio $\frac{\lambda_1}{\lambda_2}$ remains constant, then for the mean over the states (7.14) holds [Kov99]. The interpretation of this theoretical property is, that the system remains very long in each state, long enough so that in this state equilibrium of the continuous model part is reached, while still the relations of the state probabilities in steady state are not changed.

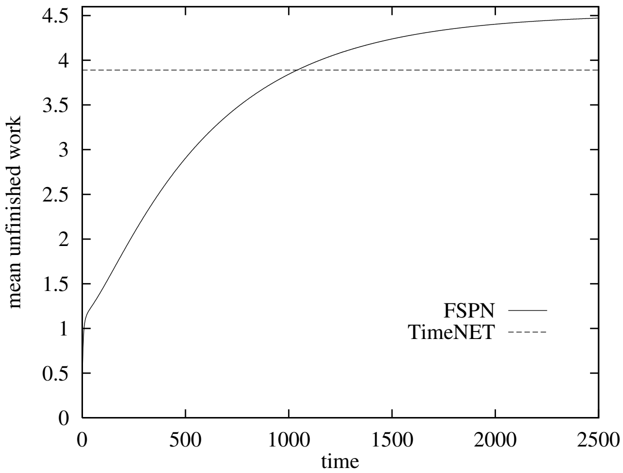

Figure 7.32: Mean unfinished work on the unrestricted domain

Therefore, the model in figure 7.29 has been changed and λ_1 and λ_2 have been scaled by the factor 1/256. This keeps the relation of λ_1 and λ_2 constant, but results in extremely small transition rates. Doing so has the disadvantage of making the models numerically less stable and of introducing an extremely long transient phase, as can be seen in the time range in the following plots.

Furthermore, stationary mean unfinished work of the DSPN model on the right side in figure 7.29 has been computed with using the stationary analysis component of TimeNET, the result being 3.89. To obtain that solution the places P_6 and P_7 have been limited to 30 customers, which gives the same result as a model with larger capacity and hence is equivalent to the unrestricted one. TimeNET can only handle models with finite state space and therefore the minimum buffer size needed, so that a larger buffer will lead to the same results, has been determined in several experiments.

The two curves in figure 7.31 come much closer to the theoretical result. Figure 7.32 shows the mean unfinished work in the corresponding unrestricted system (which means in practice $z^{min} = 0, z^{max} = 50$) and the curve crosses the stationary solution of the EDSPN, as expected.

Finally, for the model in figure 7.29 a different set of parameters is chosen. The parameters have been obtained from observations on Internet traffic with statistical methods [Gus91]. With choosing $\lambda_1 = 0.068$, $\lambda_2 = 0.0051$, $\lambda_3 = 9.61$ and $\lambda_4 = 77.37$ the MMPP arrival stream is a good model for real data traffic on the Internet. All rates are measured in sec^{-1}. In order to keep the utilization factor ρ below one, the service time distributions had to be changed into $b_1(.) = Exponential(10.0)$ and $b_2(.) = Dirac(0.01)$. This set of parameters is numerically extremely difficult to handle, therefore the stepsizes in the numerical algorithm have been set to $k = 0.0005$ for the discretization in time and $h = 0.01$ for the discretization in space. Evaluating formula (7.14) for the Internet parameters gives $av = 2.2933$.

The mean unfinished work computed over ten time units is shown in figure 7.33. The mean unfinished work after that period has just gone over 1 sec. and due to the extremely small rates λ_1 and λ_2 it will most likely take very long until the steady state value of on the average 2.29 sec. of unfinished work in the system has accumulated. Since the two modulating rates are so small, it can be assumed after the earlier observations, that in equilibrium the mean unfinished work will reach the analytical average or even go beyond it.

In summary, fluid places are a very intuitive means of modelling continuous system components as well as variables for collecting reward. The examples in this chapter have shown, that there is a number of application fields for second order FSPNs. The numerical solution component is not equally suitable for all of them. In some of the examples one can not rely on good accuracy of the results. This is mainly the case if the fluid place replaces a discrete place and when the fluid jumps are used. In modelling accumulated reward accuracy proved to be very high and the hybrid system can not be modelled otherwise, so there is no alternative model, that can be used for comparisons. To my knowledge, there is not even a simulation for second order FSPNs available.

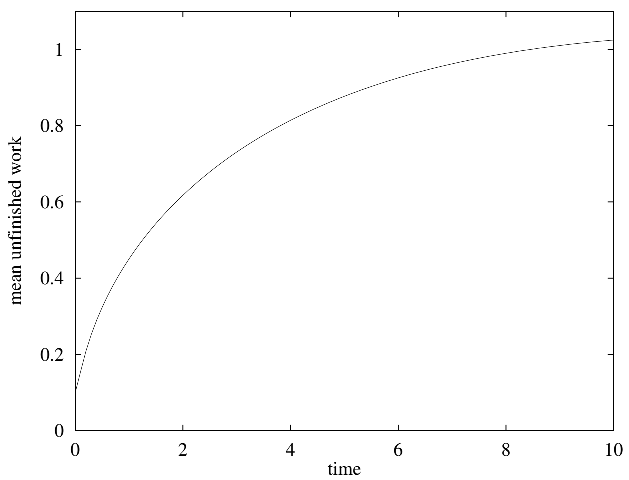

Figure 7.33: Mean unfinished work for the Internet parameters

Broader studies are necessary to be able to clearly define in which situation a second order FSPN is a good choice and where some other formalism, or a discrete SPN, should be preferred.

Chapter 8

Conclusions and Outlook

In this thesis an extension of the known FSPNs as introduced in [TK93] and [HKNT98], called *second order FSPNs* has been introduced. The formalism expands FSPNs in so far as it allows for a variability in the fluid flow, which in consequence also is a stochastic element in the model. The discrete part of second order FSPNs has been limited to the class of generalized stochastic Petri nets (GSPNs). Hence, the stochastic process underlying a second order FSPN is still a Markov process with mixed continuous and discrete state space. The differential equations describing the process' temporal behaviour are a system of second order partial differential equations of the diffusion-convection type, also called Fokker-Planck equations. Fluid queueing systems are mathematically modelled by just one of these equations, whereas underlying a fluid Petri net are as many equations as the Petri net has discrete states and they are coupled by the generator matrix.

The equation for fluid queueing systems on an unlimited domain can be solved analytically, as shown in chapter 3. This is no longer the case for fluid Petri nets where numerical solution methods must be applied. A major aim of this thesis has been to make available numerical solution methods for the transient analysis of second order FSPNs with one and two fluid places.

Second order FSPNs have been extended further by introducing jump transitions with respect to the fluid places. These transitions deposit or remove a certain amount of fluid at once, whereas the fluid flow happens in an amount per unit of time. The fluid jumps allow for an instantaneous jump up or down, that may be used for a reset of a fluid level or an immediate emptying of a fluid place.

The initial intention was to develop a modelling formalism that applies to the modelling of continuous system components, such as the reactor content and the temperature in section 7.3 and to the approximation of non-exponential distributions as in the example in section 7.1. In addition to those application fields, fluid places have proven to be a very intuitive means of directly modelling reward measures in performability modelling. An example of accumulated rate reward has been studied in section 7.2 and unfinished work in a queueing system in 7.4. For both full probability densities have been obtained, from which the means are computed. In a discrete model the two measures require additional calculations to obtain the mean values. For the full probability density analytical solutions exist only in very rare cases. If no analytical formula is available the density's moments can be estimated, which in general is not easy.

All the experiments have been carried out with the transient solution component. Stationary results, if needed, have been computed by running the transient analysis long enough, so that the model was in steady state. This may have caused some inaccuracies, that are not due to the formalism. As soon as the stationary analysis component is completed it will be applied to the approximation example in section 7.1 and this will very likely improve the results.

In order to be able to model control and regulation in manufacturing systems the dependencies between the continuous and the discrete model part have been defined very flexibly. The fluid flow rates may be different in each discrete state and also may depend on the fluid levels. The latter has been used in the modelling of the reactor in section 7.3 and it has been studied in [Zis98]. The fluid jump rates may vary in the different discrete states and along the continuous axis, whilst the jump heights are constant in each discrete state.

Not only the fluid parameters may be dependent on the continuous state, also the discrete transition rates can vary over the continuous state space. The dependency of the discrete transition rates on the continuous state is still in progress and will be used in the model of the batch reactor, where the controller changes its state if the temperature rises beyond some limit.

A serious drawback of the numerical solutions, compared to the analytical solutions obtained in queueing theory, is the lack of accuracy. The algorithms described in this thesis approximate the differential equations with order o(h,k). So, the errors are caused on one hand by the discretization and on the other hand by the fact that the continuous axis always must be bounded. It takes quite some effort to tune the algorithm parameters appropriately and to choose suitable boundary points, so that a sufficiently accurate approximation to a discrete model is obtained, as can be remarked in the study of the approximation in section 7.1.

So, in some cases fluid models allow for the modelling of systems that can not be modelled with discrete Petri nets and in some cases measures are obtained, that can not be computed with a discrete model, but the solution will be in very few cases identical with the results obtained from a discrete model. One would have to make large series of experiments to be able to predict accuracy. In general, one can say, the simpler the model, the better is the approximation.

This thesis should be seen as one among the first steps in a long row. Therefore much work remains uncompleted and many questions remain open. In the remainder of this section topics for further research will be discussed.

The stationary analysis is still in process of development. As in most stochastic processes the stationary analysis is mathematically simpler than the transient analysis. But new difficulties arise in the incorporation of the boundary conditions and in memory complexity. The coupling term can no longer be evaluated explicitly (as it has been done for the transient analysis) and therefore the discretization yields a system of linear equations that involves all discrete states at once and has thus very quickly extremely high dimension.

A way to avoid this explosion in the dimensions of the system of linear equations is to use an alternating direction implicit (ADI) method (see [Str89] pp. 142 ff.), where the full problem is split into a subproblem for each discrete state and the subproblems are solved alternatingly. The disadvantage of ADI methods is, that they are less accurate than a full explicit discretization. Whether the inaccuracy of an ADI scheme is tolerable can only be judged after it has been implemented. For the time being it remains one of the open questions.

ADI discretization schemes are also applicable to the transient analysis of models with more than one continuous dimension. The 'directions' then are the continuous dimensions. In the algorithm used for two continuous dimensions at the moment the cross-product of the two directions is built and stored in a long vector (see equation (6.110)). Decoupling the two directions would save a tremendous amount of memory and would be much faster, because less data is processed, but again it is not known whether the accuracy is sufficiently high for being used in probabilistic problems.

Parallelization of the numerical algorithms will be in several ways beneficial. In an ADI method as well as in the numerical algorithm for the transient analysis described in chapter 6 many independent subproblems must be solved. This could be done in parallel and the coefficient matrices can then be kept in memory so they do not have to be computed again in every time step. Parallelization could reduce runtime almost by factor of the number of discrete states in a model, but it needs a completely new implementation.

With efficient numerical methods a third continuous dimension might be tractable. Mainly in process engineering often more than two continuous components are present in a system and it would be helpful to be able to directly model such systems with more fluid places. With the method presented in this thesis more than two continuous dimensions can not be handled. Because of memory and run time complexity the models that could be solved in three dimensions would be only toy examples.

The fluid places have been assumed to be uncorrelated. This is a simplification, that can not always be tolerated. Therefore the algorithm shall be extended, so that correlated fluid streams can be modelled correctly. It would be helpful, if the coefficients of correlation would be computed automatically from the fluid flow rates, so that the user does not have to do this by hand.

The formalism is easily extended to non-homogeneous second order FSPNs. At the moment the parameter matrices for each discrete state are computed again in each time step. It would be little additional effort to let the parameters depend on time and solve non-homogeneous second order FSPNs.

The computation of measures has yet been reduced to the means and the marginal densities in the two dimensional models. It would, however, be useful to have all quantiles and a variable choice of sums over the discrete states. There is not much theoretical challenge in implementing a wide variety of measures, but it would certainly give more insight in the models being analyzed and it is crucial for decision taking with the help of the modelling process.

For obtaining a convenient tool the net class FSPN is just being developed for the graphical user interface AGNES, so that soon second order FSPNs will become part of TimeNET.

Bibliography

[ABCD95] M. Ajmone Marsan, G. Balbo, G. Conte, and S. Donatelli. *Modelling with Generalized Stochastic Petri Nets*. Series in Parallel Computing. John Wiley & Sons, 1995.

[AC87] M. Ajmone Marsan and G. Chiola. On Petri Nets with Deterministic and Exponentially Distributed Firing Times. In G. Rozenberg, editor, *Advances in Petri Nets*, volume 266 of *LNCS*. Springer Verlag, 1987.

[ACB84] M. Ajmone Marsan, G. Conte, and G. Balbo. A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM Transactions on Computer Systems*, **2**(2), pp. 93–122, May 1984.

[Ajm90a] M Ajmone Marsan. Stochastic Petri Nets: an elementary Introduction. In G. Rozenberg, editor, *Advances in Petri Nets 1989*, volume 424 of *Lecture Notes in Computer Science*, pp. 1–29. Springer Verlag, 1990.

[Ajm90b] M. Ajmone Marsan. Stochastic Petri Nets: an Elementary Introduction. In G. Rozenberg, editor, *Advances in Petri Nets*, volume 424 of *LNCS*, pp. 1–29. Springer Verlag, 1990.

[AMS82] D. Anick, D. Mitra, and M. Sondhi. Stochastic Theory of a Data-Handling System with Multiple Sources. *The Bell System Technical Journal*, **61**(8), pp. 1871–1894, 1982.

[Ani95] V. V. Anisimov. Diffusion Approximations in Switching Stochastic Models and Applications. In A. V. Skorokhod and Y. V. Borivskikh, editors, *Exploring Stochastic Laws*, pp. 13–40. VSP, The Netherlands, 1995.

[Ani96] V. V. Anisimov. *Matrix-Analytic Methods in Stochastic Models*, volume 183 of *Lecture Notes in Pure and Appl. Mathem. Series*, chapter Asymptotic Analysis of Switching Queueing Systems in Conditions of Low and Heavy Loading, pp. 241–260. Marcel Dekker, Inc., 1996.

[Arn95] A. Arnold. Numerically Absorbing Boundary Conditions for Quantum Evolution Equations. In *Proc. Int. Workshop on Computational Electronics*, Tempe, AZ, USA, 1995. VLSI design (1996).

[Asm95] S. Asmussen. Stationary Distributions for Fluid Flow Models with or without Brownian Noise. *Commun. Statist. – Stochastic Models*, **11**(1), pp. 21–49, 1995.

[BDF+94] G. Balbo, S. Donatelli, G. Franchescinis, A. Mazzeo, N. Mazzocca, and M. Ribaudo. On the Computation of Performance Characteristics of Concurrent Programs Using GSPNs. *Performance Evaluation*, **19**, pp. 195–222, 1994.

[Box96] O. J. Boxma. Fluid queues and regular variation. *Performance Evaluation*, **27&28**, pp. 699–712, 1996.

[BS68] I. N. Bronstein and K. A. Semendjajew. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Zürich, Frankfurt/M., 1968.

[BT86] A. Bobbio and K. S. Trivedi. An Aggregation for the Transient Analysis of Stiff Markov Chains. *IEEE Transactions on Computers*, **C-35**(9), pp. 803–814, Sept. 1986.

[Buc91] P. Buchholz. *Die strukturierte Analyse Markovscher Modelle*, volume 282. Springer-Verlag, PhD Thesis, Universität Dortmund, 1991. Informatik-Fachberichte.

[Buc92] P. Buchholz. A hierarchical view of GCSPNs and its impact on qualitative and quantitative analysis. *Parallel and Distributed Computing*, **15**, pp. 207–224, 1992.

[CBPFC+93] G. Ciardo, A. Blakemore, jr. P. F. Chimento, J. K. Muppala, and K. S. Trivedi. Automated Generation and Analysis of Markov Reward Models Using Stochastic Reward Nets. In C. Meyer and R. J. Plemmons, editors, *Linear Algebra, Markov Chains and Queueing Models*, volume 48 of *IMA Volumes in Mathematics and it's Applications*. Springer Verlag, 1993.

[CFJ+91] J.A. Couvillon, R. Freire, R. Johnson, W.D. Obal II, M.A. Qureshi, M. Rai, W.H. Sanders, and J.E. Tvedt. Performability Modeling with UltraSAN. *IEEE Software*, pp. 69–80, September 1991.

[Chi92] G. Chiola. GreatSPN 1.5 software architecture. In G. Balbo and G. Serazzi, editors, *Proc 5th Int. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation*, Computer Performance Evaluation. Modelling Techniques and Tools, pp. 117–132, Torino, Italy, 13-15 February 1991 1992. Elsevier, Amsterdam.

[Cin75] E. Cinlar. *Introduction to Stochastic Processes*. Prentice Hall, Englewood Cliffs, 1975.

[CKT94] H. Choi, V. G. Kulkarni, and K. S. Trivedi. Markov regenerative stochastic Petri nets. *Performance Evaluation*, **20**, pp. 337–357, 1994.

[CM65] D.R. Cox and H.D. Miller. *The Theory of Stochastic Processes.* Chapman and Hall, 1965.

[CNT99] G. Ciardo, D. Nicol, and K.S. Trivedi. Discrete-event simulation of fluid stochastic petri nets. *IEEE Trans. Softw. Eng.*, **25**(2), pp. 207–217, March/April 1999.

[Cox55] D. R. Cox. The Analysis of Non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables. *Proc. Camb. Phil. Soc. (Math. and Phys. Sciences)*, **51**, pp. 433–441, 1955.

[Cox84] D. R. Cox. Long-range Dependence: A Review. In H. A. David and H. T. David, editors, *Statistics an Appraisal, Proc. 50th Anniversary Conf. Iowa State Statistical Lab.*, pp. 55–74. Iowa State University Press, 1984.

[CS95] T. M. Chen and V. K. Samalam. Time-dependent behavior of fluid buffer models with Markov input and constant output rates. *SIAM Journal on Applied Mathematics*, **55**(3), pp. 784–799, June 1995.

[CT93] G. Ciardo and K. S. Trivedi. A Decomposition Approach for Stochastic Reward Net Models. *Performance Evaluation*, **18**(1), pp. 37–60, July 1993.

[DA92] R. David and H. Alla. *Petri Nets and Grafcet.* Prentice Hall, 1992.

[DA93] R. David and H. Alla. Autonomous and Timed Continuous Petri nets. In G. Rozenberg, editor, *Advances in Petri Nets 1993*, volume 674 of *Lecture Notes in Computer Science*, pp. 71–90. Springer-Verlag, 1993.

[DK98] I. Demongodin and N. T. Koussoulas. Modelling of hybrid control Systems using Petri nets. In J. Zaytoon, editor, *Proc. 3ème Conf. Int. sur l'Automatisation des Processus Mixtes*, Les Systèmes Dynamiques Hybrides (Hybrid Dynamical Systems), pp. 330–337, Reims, March 1998.

[Dra98] R. Drath. Hybrid Object Nets: An Object Oriented Concept for Modelling Complex Hybrid Systems. In J. Zaytoon, editor, *Proc. 3ème Conf. Int. sur l'Automatisation des Processus Mixtes*, Les Systèmes Dynamiques Hybrides (Hybrid Dynamical Systems), pp. 436–442, Reims, March 1998.

[DRH95] S. Donatelli, M. Ribaudo, and J. Hillston. A Comparison of Performance Evaluation Process Algebra and Generalized Stochastic Petri Nets. In *Proc. Sixth International Workshop on Petri Nets and Performance Models - PNPM'95*, pp. 158–168, Durham, N. Carolina, USA, Oct. 3–6 1995. IEEE-CS Press.

[DS98] G. Decknatel and E. Schnieder. Modelling Railway Systems with Hybrid Petri Nets. In J. Zaytoon, editor, *Proc. 3ème Conf. Int. sur l'Automatisation des Processus Mixtes*, Les Systèmes Dynamiques Hybrides (Hybrid Dynamical Systems), pp. 309–315, Reims, March 1998.

[dSG92] E. de Souza e Silva and H. R. Gail. Performability analysis of computer systems: from model specification to solution. *Performance Evaluation*, **14**, pp. 157–196, 1992.

[Dud86] A. Duda. Diffusion Approximation for Time-Dependent Queueing Systems. *IEEE J. Sel. Areas in Commun.*, **SAC-4, 6**, pp. 905–918, 1986.

[EM94] A. I. Elwalid and D. Mitra. Statistical Multiplexing with Loss Priorities in Rate-Based Congestion Control of High-Speed Networks. *IEEE trans. on comm.*, **42**(11), pp. 2989–3002, November 1994.

[Fel54] Feller. Diffusion Processes in one dimension. *Transactions of the American Mathematical Society*, **77**, pp. 1–31, 1954.

[Fel68a] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume I. John Wiley & Sons, Inc., 1968.

[Fel68b] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume II. John Wiley & Sons, Inc., 1968.

[Fic64] G. M. Fichtenholz. *Differential- und Integralrechnung I.* VEB Deutscher Verlag der Wissenschaften, 1964.

[Fla98] J. M. Flaus. Hybrid Supervisor Synthesis for a class of Hybrid Systems. In J. Zaytoon, editor, *Proc. 3ème Conf. Int. sur l'Automatisation des Processus Mixtes*, Les Systèmes Dynamiques Hybrides (Hybrid Dynamical Systems), pp. 436–442, Reims, March 1998.

[Gav71] D. P. Gaver. Analysis of Remote Terminal Backlogs under Heavy Demand Conditions. *Journal of the ACM*, **18**(3), pp. 405–415, July 1071.

[Gav68] D. P. Gaver. Diffusion Approximations and Models for certain Congestion Problems. *J. Appl. Prob.*, **5**, pp. 607–623, 1968.

[Gel75] E. Gelenbe. On Approximate Computer System Models. *Journal of the ACM*, **22**(2), pp. 261–269, April 1975.

[Gel79] E. Gelenbe. Probabilistic Models of Computer Systems. Part II: Diffusion Approximations, Waiting Times and Batch Arrivals. *ACTA INFORMATICA*, **12**, pp. 285–303, 1979.

[Ger94] R. German. *Analysis of Stochastic Petri Nets with non-Exponentially Distributed Firing times.* PhD thesis, Technische Universität Berlin, March 1994.

[GKZH95] R. German, C. Kelling, A. Zimmermann, and G. Hommel. TimeNET: a toolkit for evaluating non-Markovian stochastic Petri nets. *Performance Evaluation*, **24**, pp. 69–87, 1995.

[GL79] D. P. Gaver and J. P. Lehoczky. A diffusion approximation model for a communication system allowing message interference. *IEEE trans. on commun.*, **COM-27**, pp. 1190–1199, 1979.

[GL81] H. J. Genrich and K. Lautenbach. System Modelling with High-Level Petri Nets. *Theoretical Computer Science*, **13**(1), pp. 109–136, January 1981.

[GL94] R. German and C. Lindemann. Analysis of stochastic Petri nets by the method of supplementary variables. *Performance Evaluation*, **20**, pp. 317–335, 1994.

[GM84] D. Gross and D. R. Miller. The Randomization Technique as a Modelling Tool and Solution Procedure for Transient Markov Processes. *Operations Research*, **32**, pp. 345–361, 1984.

[GMO96] E. Gelenbe, X. Mang, and R. Onvural. Diffusion based statistical call admission control in ATM. *Performance Evaluation*, **27&28**, pp. 411–436, 1996.

[Gne62] B. V. Gnedenko. *The Theory of Probability.* Chelsea Publishing Company, New York, 1962.

[GP76] E. Gelenbe and G. Pujolle. The Behaviour of a Single Queue in a General Queueing Network. *ACTA INFORMATICA*, **7**, pp. 123–136, 1976.

[Gra77] W. K. Grassmann. Transient Solutions in Markovian Queueing Systems. *Comput. & Operations Research*, **4**(1), pp. 47–53, 1977.

[GU98] A. Guia and E. Usai. Modelling hybrid Systems by high-level Petri Nets. In J. Zaytoon, editor, *Proc. 3ème Conf. Int. sur l'Automatisation des Processus Mixtes*, Les Systèmes Dynamiques Hybrides (Hybrid Dynamical Systems), pp. 316–323, Reims, March 1998.

[Gus91] R. Gusella. Characterizing the Variability of Arrival Processes with Indexes of Dispersion. *IEEE Journal on Selected Areas in Communication*, **9**(2), pp. 203–211, February 1991.

[Hal72] S. Halfin. Steady-State Distribution for the Buffer Content of an M/G/1 Queue with Varying Service Rate. *SIAM J. Appl. Math.*, **23**(3), pp. 356–363, November 1972.

[Hav93] B. R. Haverkort. Approximate performability and dependability analysis using generalized stochastic Petri nets. *Performance Evaluation*, **18**(1), pp. 61–78, July 1993.

[Hav95] B. R. Haverkort. Approximate Analysis of Networks of PH/PH/1/K Queues: Theory and Tool Support. In H. Beilner and F. Bause, editors, *Quantitative Evaluation of Computing and Communication Systems*, volume 977 of *Lecture Notes in Computer Science*, pp. 239–253. Springer, 1995.

[Hey75] D. P. Heyman. A Diffusion Model Approximation for the GI/G/1 Queue in Heavy Traffic. *The Bell System Technical Journal*, **54, 9**, 1975.

[HF77] B. Halachmi and W. R. Franta. A Diffusion Approximate Solution to the G/G/k Queueing System. *Comput. & Ops. Res., Vol. 4, :37-46*, 1977.

[HF78] B. Halachmi and W. R. Franta. A Diffusion Approximation to the Multi-Server Queue. *Management Science, Vol. 24, :5, 522-529*, 1978.

[HG97] A. Heindl and R. German. A Fourth-Order Algorithm with Automatic Stepsize Control for the Transient Analysis of DSPNs. In *Proc. Seventh Int. Workshop on Petri Nets and Performance Models*, pp. 60–69, Saint Malo, France, 3–6 June 1997. IEEE-CS.

[HKNT98] G. Horton, V. G. Kulkarni, D. M. Nicol, and K. S. Trivedi. Fluid Stochastic Petri Nets: Theory, Applications and Solution. *European Journal of Operations Research*, **105**(1), pp. 184–201, February 1998.

[HL86] H. Heffes and D. M. Lucantoni. A Markov Modulated Charakterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance. *IEEE Journ. on Sel. Areas in Commun.*, **SAC-4**(6), pp. 856–868, 1986.

[HO97] B. R. Haverkort and A. Ost. Steady-State Analysis of Infinite Stochastic Petri Nets: Comparing the Spectral Expansion and the Matrix-Geometric Method. In *Proc. Seventh International Workshop on Petri Nets and Performance Models - PNPM'97*, pp. 36–45, Saint Malo, France, June 3–6 1997. IEEE-CS Press.

[Hor96] G. Horton. Computation of the Distribution of Accumulated Reward with Fluid Stochastic Petri nets. In *Proc. 2nd IEEE Int. Computer Performance and Dependability Symposium (IPDS)*, pp. 90–95, Urbana-Champaign, Illinois, USA, 4-6 September 1996. IEEE-CS Press.

[Hor98] G. Horton. Fluide stochastische Petri-Netze. Habilitation, Friedrich-Alexander Universität Erlangen, July 1998.

[HS72] S. Halfin and M. Segal. A Priority Queueing Model for a Mixture of Two Types of Customers. *SIAM J. Appl. Math.*, **23**(3), pp. 369–379, November 1972.

[Jen53] A. Jensen. Markoff Chains as an Aid in the Study of Markoff Processes. *Skand. Aktuarietidskrift*, **36**, pp. 87–91, 1953.

[Jen92] K. Jensen. *Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use, Vol. 1: Basic Concepts.* EATCS Monographs in Theoretical Computer Science. Springer-Verlag, 1992.

[JR91] K. Jensen and G. Rozenberg, editors. *High-level Petri Nets.* Springer Verlag, 1991.

[Kar66] S. Karlin. *A First Course in Stochastic Processes.* Academic Press, New York, 1966.

[KHR89] S. V. Krishnamachari, L. J. Hayes, and T. F. Russel. A Finite Element Alernating-Direction Method Combined with a Modified Method of Characteristics for Convection-Diffusion Problems. *SIAM Journal on Numerical Analysis*, **26**(6), pp. 1462–1473, December 1989.

[Kim83] T. Kimura. Diffusion Approximation for an M/G/m Queue. *Operations Research*, **31**(2), pp. 304–321, 1983.

[Kin62] J. F. C. Kingman. On queues in heavy traffic. *J. Royal Statist. Soc.*, **Series B**(24), pp. 383–392, 1962.

[KK95] R. L. Karandikar and V. G. Kulkarni. Second-Order Fluid Flow Models: Reflected Brownian Motion in a Random Environment. *Operations Research*, **43**(1), pp. 77–88, Jan.-Feb. 1995.

[Kle75] L. Kleinrock. *Queueing Systems Vol I: Theory.* John Wiley, 1975.

[Kle76] L. Kleinrock. *Queueing Systems Vol II: Computer Applications.* John Wiley, 1976.

[Kne91] C. Knessl. On the Diffusion Approximation to a Fork and Join Queueing Model. *SIAM Journal on Applied Mathematics*, **51**(1), pp. 160–171, February 1991.

[Kob74a] H. Kobayashi. Application of the Diffusion Approximation to Queueing Networks I: Equilibrium Queue Distributions. *Journal of the ACM*, **21**(2), pp. 316–328, April 1974.

[Kob74b] H. Kobayashi. Application of the Diffusion Approximation to Queueing Networks II: Nonequilibrium Distributions and Applications to Computer Modeling. *Journal of the ACM*, **21**(3), pp. 459–469, July 1974.

[KOH77] H. Kobayashi, Y. Onozato, and D. Huynh. An approximate method for design and analysis of an ALOHA-system. *IEEE trans. on commun.*, **COM-25**, pp. 148–157, 1977.

[Kov99] I. Kovalenko. Personal communication, 1999.

[KT81] S. Karlin and H. M. Taylor. *A Second Course in Stochastic Processes.* Academic Press, New York, 1981.

[KT86] G. Kimura and Y. Takahashi. Diffusion Approximation for a Token Ring System with Nonexhaustive Service. *IEEE J. on Sel. Areas on Commun.*, **4, 6**, pp. 794–801, 1986.

[Kue79] P. J. Kuehn. Approximate Analysis of General Queueing Networks by Decomposition. *IEEE Transactions on Communications*, **COM-27**(1), pp. 113–126, January 1979.

[Lap92] J. C. Laprie, editor. *Dependability: Basic Concepts and Terminology*, volume 5 of *Dependable Computing and Fault-Tolerant Systems.* Springer, Vienna, 1992.

[LS79] P. A. W. Lewis and G. S. Shedler. Simulation of Nonhomogeneous Poisson Processes by Thinning. *Naval Research Logistics Quarterly*, **26**(3), pp. 403–413, 1979.

[LS96] C. Lindemann and G. Shedler. Numerical Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions. *Performance Evaluation*, **27**, pp. 565–582, 1996. Special Issue Proc. of PERFORMANCE '96.

[LS99] U. Leuthäusser and J. Sellen. Reales Ausfallverhalten eines Gerätebestandes. *at-Automatisierungstechnik*, **3**, pp. 110–117, March 1999.

[LTWW94] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM trans. on networking*, **2**(1), pp. 1–15, February 1994.

[Mac97] M. Maczek. Entwurf und Implementierung eines Simulators für Fluide, Stochastische Petri-Netze. Master's thesis, Technische Universität Berlin, 1997.

[MC95] I. Mitrani and R. Chakka. Spectral expansion solution for a class of Markov models: application and comparison with the matrix-geometric method. *Performance Evaluation*, **23**, pp. 241–260, 1995.

[Mey80] J. F. Meyer. On Evaluating the Performability of Degradable Computing Systems. *IEEE Trans. on Computers*, **C-29**(8), pp. 720–731, August 1980.

[Mey82] J. F. Meyer. Closed-Form Solutions of Performability. *IEEE Transactions on Computers*, **C-31**(7), pp. 648–657, July 1982.

[MFW80] J. F. Meyer, D. G. Furchtgott, and L. T. Wu. Performability Evaluation of the SIFT Computer. *IEEE Trans. on Computers*, **C-29**(6), pp. 501–509, June 1980.

[Mit88] D. Mitra. Stochastic Theory of a Fluid Model of Producers and Consumers Coupled by a Buffer. *Advances in Applied Probability*, **20**, pp. 646–676, 1988.

[MM94] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations.* Cambridge University Press, 1994.

[MMT94] J. Muppala, M. Malhotra, and K. S. Trivedi. Stiffness-Tolerant Methods for Transient Analysis of Stiff Markov Chains. *Microelectronics and Reliability*, **34**(11), pp. 1825–1841, 1994.

[Mol82] M. K. Molloy. Performance Analysis Using Stochastic Petri Nets. *IEEE Trans. on Computers*, **C-31**(9), pp. 913–917, September 1982.

[Mol85] M. K. Molloy. Discrete Time Stochastic Petri Nets. *IEEE Trans. on Software Engineering*, **SE-11**(4), pp. 417–423, April 1985.

[Mor93] J. A. Morrison. Diffusion Approximation for Head-of-the-Line Processor Sharing for Two Parallel Queues. *SIAM Journal on Applied Mathematics*, **53**(2), pp. 471–490, April 1993.

[MT95] V. Mainkar and K. S. Trivedi. Fixed Point Iteration Using Stochastic Reward Nets. In *Proc. Sixth International Workshop on Petri Nets and Performance Models - PNPM'95*, pp. 21–30, Durham, N. Carolina, USA, October 3–6 1995. IEEE-CS Press.

[Mur89] T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, **77**(4), pp. 541–581, April 1989.

[NA80] Y. W. Ng and A. A. Avizienis. A Unified Reliability Model for Fault-Tolerant Computers. *IEEE Trans. on Computers*, **C-29**(11), pp. 1002–1011, Nov. 1980.

[Neu94] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models.* Dover Publications Inc., 1994.

[New65] G. F. Newell. Approximation Methods for Queues with Application to the Fixed-Cycle Traffic Light. *Siam Review*, **7**(2), pp. 223–240, April 1965.

[New68a] G. F. Newell. Queues with Time-Dependent Arrival Rates I–the Transition through Saturation. *J. Appl. Prob.*, **5**, pp. 436–451, 1968.

[New68b] G. F. Newell. Queues with Time-Dependent Arrival Rates II–the Maximum Queue and the Return to Equilibrium. *J. Appl. Prob.*, **5**, pp. 579–590, 1968.

[New68c] G. F. Newell. Queues with Time-Dependent Arrival Rates III–a Mild Rush-Hour. *J. Appl. Prob.*, **5**, pp. 591–606, 1968.

[New71] G. F. Newell. *Applications of Queueing Theory.* Chapman and Hall, Ltd. (London), 1971.

[Ngu95] V. Nguyen. Fluid and Diffusion Approximations of a Two-Station Mixed Queueing Network. *Mathematics of Operations Research*, **20**(2), pp. 321–354, May 1995.

[NM95] D. M. Nicol and A. S. Miner. The Fluid Stochastic Petri Net Simulator. In *Proc. Sixth International Workshop on Petri Nets and Performance Models - PNPM'95*, pp. 214–215, Durham, North Carolina, USA, October 1995. IEEE-CS Press.

[Onv94] R. O. Onvural. *Asynchronous Transfer Mode Networks: Performance Issues.* Artech House, 1994.

[Pet62] C. A. Petri. *Kommunikation mit Automaten.* PhD thesis, Universität Bonn, Bonn, Germany, 1962.

[QSvMG96] M. A. Qureshi, W. H. Sanders, A. P. A. van Moorsel, and R. German. Algorithms for the Generation of state-level Representations of Stochastic Activity Networks with General Reward Structures. *IEEE Transactions on Software Engineering*, **22**(9), pp. 603–614, September 1996.

[Rei85] W. Reisig. *Petri nets.* Springer Verlag, Berlin, 1985.

[RH80] C. V. Ramamoorthy and G. S. Ho. Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets. *IEEE Transactions on Software Engineering*, **SE-6**(5), pp. 440–449, September 1980.

[RK74] M. Reiser and H. Kobayashi. Accuracy of the Diffusion Approximation for some Queuing Systems. *IBM Journal of Research and Development*, **18**(2), pp. 110–124, March 1974.

[RT88] A. Reibman and K. S. Trivedi. Numerical Transient Analysis of Markov Models. *Computers and Operations Research*, **15**(1), pp. 19–36, 1988.

[Sch92] E. Schnieder, editor. *Petrinetze in der Automatisierungstechnik.* B. G. Teubner Verlag, Stuttgart, 1992.

[SDQW95] W. S. Sanders, W. D. Obal II, M. A. Qureshi, and F. W. Widjanarko. The *UltraSAN* Modeling Environment. *Performance Evaluation*, **24**(1), pp. 89–115, Oct./Nov. 1995.

[Sim91] A. Simonian. Stationary Analysis of a Fluid Queue with Input Rate Varying as an Ornstein-Uhlenbeck Process. *SIAM Journal on Applied Mathematics*, **51**(3), pp. 828–842, June 1991.

[SKPT98] O. Stursberg, S. Kowalewski, J. Preussig, and H. Treseler. Block-Diagram Based Modelling and Analysis of Continuous Processes under Discrete Control. In J. Zaytoon, editor, *Proc. 3ème Conf. Int. sur l'Automatisation des Processus Mixtes*, Les Systèmes Dynamiques Hybrides (Hybrid Dynamical Systems), pp. 63–70, Reims, March 1998.

[SM91] W. H. Sanders and J. F. Meyer. A Unified Approach for Specifying Measures of Performance, Dependability, and Performability. In A. Avizienis and J. Laprie, editors, *Dependable Computing for Critical Applications*, volume 4 of *Dependable Computing and Fault-Tolerant Systems*, pp. 215–237. Springer, 1991.

[Som49] A. Sommerfeld. *Partial Differential Equations in Physics.* Academic Press, New York, 1949.

[Sta90] P. H. Starke. *Analyse von Petrinetzen.* B. G. Teubner Verlag, Stuttgart, 1990.

[Sto94] J. Stoer. *Numerische Mathematik 1.* Springer Verlag, Berlin, 7 edition, 1994.

[STP95] R. A. Sahner, K. S. Trivedi, and A. Puliafito. *Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package.* Kluwer Academic Publishers, 1995.

[STR88] R. M. Smith, K. S. Trivedi, and A. V. Ramesh. Performability Analysis: Measures, an Algorithm, and a Case Study. *IEEE Transactions on Computers*, **37**(4), pp. 406–417, April 1988.

[Str89] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations.* Wadsworth & Brooks/Cole, 1989.

[TB95] M. Telek and A. Bobbio. Markov Regenerative Stochastic Petri Nets with Age Type General Transitions. In G. De Michelis and M. Diaz, editors, *Application and Theory of Petri NEts*, volume 935 of *LNCS*. Springer Verlag, 1995.

[TBJ+95] M. Telek, A. Bobbio, L. Jereb, A. Puliafito, and K. S. Trivedi. Steady State Analysis of Markov Regenerative SPN with Age Memory Policy. In F. Bause H. Beilner, editor, *Quantitative Evaluation of Computing and Communication Systems*, number 977 in LNCS, pp. 165–179. Springer, 1995.

[TK93] K. S. Trivedi and V. G. Kulkarni. FSPNs: Fluid Stochastic Petri Nets. In *Proc. 14th Int. Conf. on the Application and Theory of Petri Nets*, pp. 24–31, Chicago, 1993.

[vD90] N. M. van Dijk. On a Simple Proof of Uniformization for Continuous and Discrete-State Continuous-Time Markov Chains. *Advances in Applied Probability*, **22**, pp. 749–750, 1990. letter to the editor.

[vM94] A. P. A. van Moorsel. *Performability Evaluation. Concepts and Techniques.* PhD thesis, Twente University, The Netherlands, 1994.

[vMS97] A. P. A. van Moorsel and W. H. Sanders. Transient Solution of Markov Models by Combining Adaptive & Standard Uniformization. *IEEE Transactions on Reliability*, **46**(3), pp. 430–440, September 1997.

[WG96] K. Wolter and R. German. Second Order Non-Markovian Fluid Stochastic Petri Nets. In *Proc. Workshop on Perf. Modeling of Comp. and Comm. Systems (PMCCS3)*, Bloomingdale, IL, USA, September 1996.

[WH97] K. Wolter and G. Hommel. Hybrid Modelling with Second Order Fluid Stochastic Petri Nets. In *Proc. Workshop on Parallel and Distr. Real-Time Systems*, pp. 239–243, Geneva, Switzerland, April 1–3 1997. IEEE-CS.

[WHG96] K. Wolter, G. Horton, and R. German. Non-Markovian Fluid Stochastic Petri Nets. Technical Report 96-13, TU Berlin, 1996.

[Whi83] W. Whitt. The Queueing Network Analyzer. *The Bell System Technical Journal*, **62**(9), pp. 2779–2815, November 1983.

[Wol97] K. Wolter. Second Order Fluid Stochastic Petri Nets: an Extension of GSPNs for Approximate and Continuous modelling. In *Proc. 1st World Congress on Systems Simulation (WCSS'97)*, pp. 328–332, Singapore, Sept. 1–3 1997.

[Wol98a] K. Wolter. Probabilistic and Deterministic fluid flow in second order FSPNs. In J. Zaytoon, editor, *Proc. 3ème Conf. Int. sur l'Automatisation des Processus Mixtes*, Les Systèmes Dynamiques Hybrides (Hybrid Dynamical Systems), pp. 324–329, Reims, March 1998.

[Wol98b] K. Wolter. Virtual Waiting time in Queues obtained from Second Order FSPNs. In *Proc. Third International Computer Performance and Dependability Symposium - IPDS'98*, p. 273, Durham, NC, USA, Sept. 7–9 1998. IEEE-CS Press.

[Wol99] K. Wolter. Jump Transitions in Second Order FSPNs. In *Proc. 7th Int. Symp. on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'99)*, pp. 156–163, Washington DC, USA, October 1999. IEEE-CS Press.

[WZ99] K. Wolter and A. Zisowsky. On Markov Reward Modelling with Second Order FSPNs. In J. T. Bradley and N. J. Davies, editors, *Proc. 15th annual UK Performance Engineering Workshop*, pp. 167–177, Bristol, UK, July 22-23 1999. TR CSTR-99-007.

[Zij97] R. Zijal. *Analysis of Discrete Time Deterministic and Stochastic Petri Nets.* PhD thesis, Technische Universität Berlin, 1997.

[Zim97] A. Zimmermann. *Modellierung und Bewertung von Fertigungssystemen mit Petri-Netzen.* PhD thesis, Technische Universität Berlin, 1997.

[Zis98] A. Zisowsky. Entwurf und Implementierung eines Verfahrens für die transiente Analyse fluider stochastischer Petri-Netze. Master's thesis, TU Berlin, August 1998.

Glossary

At this place the notation is explained. The page numbers either indicate, where the term is defined, of where it has been mentioned for the first time.

a_j	Short form for the factor in the discretization (1D)	70
ATM	Asynchronous transfer mode	25
$b(.)$	Jump height distribution	35
b_j	Short form for the factor in the discretization (1D)	70
c_l	Short form for the factor in the discretization (1D)	88
$\mathbf{B}$	Matrix of jump height distributions	37
δ	Dirac impulse	58
$diag(\mathbf{v})$	Diagonal matrix with entries $\mathbf{v}$	35
DSPN	Deterministic and stochastic Petri net	6
force_jump	Boundary strategy for the jumps	43
FSPN	Fluid stochastic Petri net	1
GSPN	Generalized stochastic Petri net	5
h	Stepsize in space (1D)	68
h_1	Stepsize in direction z_1 (2D)	78
h_2	Stepsize in direction z_2 (2D)	78
initvar	Parameter for spreading the initial condition	89
k	Stepsize in time	68
$\mathbf{\Lambda}$	Matrix of jump height distributions	36
$\check{L}$	Number of jumps down	36
$\hat{L}$	Number of jumps up	36
M	Discretization steps in direction z_1	79
$\mathbf{M}_k(z)$	Matrix of means of the normal distribution	35
MMPP	Markov modulated Poisson process	26
MRSPN	Markov regenerative stochastic Petri net	6
μ	Mean of the fluid flow	19
$\mu_{s,j}$	Discrete form of the mean in state s (1D)	69
$\mu_{s,k,j}$	Discrete form of the mean in state s for fluid place k (2D)	79
N	Number of discretization steps in space (1D)	68
N	Number of discretization steps in direction z_2 (2D)	79

$o(.)$	Landau symbol	48
Ω	Definition domain in space	49
$\mathring{\Omega}$	Inner area of the definition domain in space	50
ODE	Ordinary differential equation	14
$\boldsymbol{\pi}$	Probability vector	14
π_s	Probability for state s	68
preserve	Boundary strategy for the jumps	43
Q	Generator matrix	14
$\mathbf{Q}(\mathbf{z})$	Generator matrix in the FSPN	32
r	$\frac{k}{h^2}$	70
r_1	$\frac{k}{h_1^2}$	81
r_2	$\frac{k}{h_2^2}$	81
$R(k, s)$	Fluid change rate	34
ρ	Utilization factor	22
ρ	Upwind parameter in the discretization scheme	69
SRN	Stochastic Reward net	5
s	Discrete state	47
scv	Squared coefficient of variation	22
σ^2	Variance of the fluid flow 19	
$\sigma^2_{s,j}$	Discrete form of the variance in state s (1D)	69
$\sigma^2_{s,k,j}$	Discrete form of the variance in state s for fluid place k (2D)	79
$\boldsymbol{\Sigma}^2(z)$	Matrix of variances	35
θ	Parameter for implicity	68
τ	Parameter for the discretization of the coupling term	69
$\overset{n}{\underset{n-1}{\textcircled{\tau}}}$	Discrete form of $\boldsymbol{\pi}(t, \mathbf{z})\mathbf{Q}$	69
$u^n_{s,j}$	Discrete approximation of $\pi_s(kn, jh)$	68
$\mathbf{v}^T$	Transposed vector	72

Index