Numerical analysis of Markovian fluid queues

Author:
Miklós Vécsei

Supervisor:
Dr. Miklós Telek

Contents

Chapter 1

Introduction

Markovian fluid queues are stochastic models with a wide range of applications. Although the model can easily be understood, the numerically stable calculation of it's properties is a complicated task. First this paper compares two stable approaches, according to their speed and stability. Then it presents a method for the analysis of RED processes.

The model based analysis of telecommunication networks is an important part of the dimensioning of network devices. A widely applied model is the queuing system, which is based on probabilistic assumptions. For these systems we divide the network processes into two categories. The first is the arriving of the demands, which increases the load of the server. The second is the servicing of these demands. Both types of processes can have certain properties, (for example memoryless or deterministic property,) which can be considered in the system.

If the buffer has a large capacity, one can take the continuous limit of these models. In this limit the demands are not separated from each other, the load of the system is handled as if it was a continuous material. This models can be imagined the following way: Let us take a container, (the buffer,) which is filled with fluid to a certain height. Because of the arrival and the service of the demands, the height of the fluid may change with time, as the load on the network varies. The rate of the rising of the fluid-level is determined by an environmental process. The fluid level will rise if the environment is in a state where the arrival of the demands is faster than the serving process. Similarly if the serving of the demands is faster, than the fluid level will decrease. Usually we presume that the environmental process is memoryless. (The future states of the system depends on it's present state, but not on it's past.) In this case the environmental process can be described with a Markov process. These models are called Markovian-fluid models.

The first sections of this paper discusses the theory of these models. For large buffer-limits the differential equations needed to solve the model may get unstable. The second part of the paper will focus on this problem, and the algorithms developed to solve them. In order to examine their numerical properties I implemented the methods in MATLAB. In the last section the results will be applied for the analysis of RED processes.

To make the notations used throughout the paper clear, they are collected below:

Notation

$Z(t)$, S, Q

As it is mentioned before, we assume, that the environmental process defining the rate with which the fluid level varies, is a Markov-process. This process is only time-depended, it will be denoted by $Z(t)$. It is defined over the state space denoted by S, which is considered to be finite. The generator-matrix of the process is Q. The jith element of Q will be denoted by Q_{ji}.

$X(t)$, R

$X(t)$ notates the level of the fluid in the buffer. It varies with the rate defined by $Z(t)$. A diagonal matrix can be generated which will represent the rate for all states of the $Z(t)$ process. This matrix will be referred to as the drift matrix, and will be denoted with R: $R(i,i) = r(Z(t) = i) = \frac{dX}{dt}|_{Z(t)=i}\ \forall i \in S$

B

B denotes the capacity of the system. In several occasions it can be considered infinite.

$F(t,x,i)$, $f(t,x,i)$

$F(t,x,i)$ is the fluid-distribution of the process at time t:
$F(t,x,i) = P(X(t) \leq x, \quad Z(t) = i)$
$f(t,x,i)$ is the probability density function:
$f(t,x,i) = dF(t,x,i)/dx$

Chapter 2

The model, spectral solution

2.1 The equations for F(t,x,i)

In this section we will be summarizing the theorems of the model. First of all, when calculating the fluid rates one needs to consider that the fluid level can not be negative, neither can it exceed the buffer-level:

$$\tfrac{dX(t)}{dt} = \begin{cases} r(Z(t))^{+} & if \quad X(t) = 0 \\ r(Z(t)) & if \quad 0 < X(t) < B \\ r(Z(t))^{-} & if \quad X(t) = B \end{cases}$$

Where $f^{+} = max(f, 0)$ and $f^{-} = min(f, 0)$. The meaning of the first line of the formula is the following: When the buffer is empty it means that there are no demands in the system. At this level if the system is in a state where the rate is negative, than if a new demand arrives, it will be served immediately. For these states of the Markov process, as the fluid level can not decrease below zero, the rate is equal to 0. The last line has a similar meaning, it considers, that fluid level can not be higher, than the capacity of the system.

The process Z(t) is considered to be an irreducible continuous-time Markov chain. Therefore it has a stationary distribution:

$$\pi_j = \lim_{t \to \infty} P(Z(t) = j)$$

Let us define a vector, which consists of the π_j probabilities: $\pi = (\pi_1, \pi_2, ..., \pi_n)$. From the theory of continuous-time Markov chains follows that π satisfies the following equation:

$$\pi Q = 0$$

As all properties of the model can be calculated from $F(t, x, i)$, our goal is to obtain this function. Besides calculating the expected value and the deviation of the fluid level, $F(t, x, i)$ can also be used to compute the efficiency of the system, as it will be shown in the last section of the paper.

In [7] an equation was deducted, that we can use to compute the fluid-distribution. It can be proved, that the vector $F(t, x) = [F(t, x, 1), F(t, x, 2), ..., F(t, x, n)]$ satisfies a differential-equation, which has a form similar to the continuity-equations:

$$\frac{\partial F(t,x)}{\partial t} + \frac{\partial F(t,x)}{\partial x} R = F(t,x)Q \tag{2.1}$$

The boundary conditions of (2.1):

$$F(t,0,i) = 0 \quad if \quad r(i) > 0 \qquad\qquad F(t,B,i) = \pi_i \quad if \quad r(i) < 0$$

The first boundary condition represents, that for states with positive drift, the buffer can not be empty. (If the environmental process switches to this state at $x = 0$ the fluid level will immediately rise above the $x = 0$ height.) The second boundary condition has a similar meaning. Because the fluid level can not exceed the buffer-limit, it's height can not be B for states with negative drift.

Usually one is interested in the stationary probability distribution: $F(x) = \lim_{t\to\infty} F(t,x)$. (Assuming that with time the system advances towards a stationary solution.) For these functions equation (2.1) takes the following form:

$$\frac{\partial F(x)}{\partial x} R = F(x) Q \tag{2.2}$$

2.2 Spectral solution

A logical approach to the problem would be searching for the solution of equation (2.2) using it's spectral representation:

$$F(x) = e^{\lambda x}\Gamma$$

λ is a scalar and Γ is a row vector of length n. (n is the cardinality of the state-space of the environmental process.) Inserting this representation into equation (2.1) it takes the following form:

$$\lambda e^{\lambda x}\Gamma R = e^{\lambda x}\Gamma Q \tag{2.3}$$

Which can only be solved, when $det(\lambda R - Q) = 0$. The equation led to a generalized eigenvalue-problem. T he eigenvalues and the eigenvectors will be denoted by λ_i and Γ_i. After solving the problem, one can write $F(x)$ in the following form:

$$F(x) = \sum_i a_i e^{\lambda_i x}\Gamma_i$$

The a_i parameters of $F(x)$ can be attained by fitting the parameters to the boundary conditions. These will result in linear equations for the a_i parameters. At the margins the linear equations take the following form:

$$\begin{array}{ll} \text{At } x = 0\text{:} & \sum_i^n a_i\Gamma_{ij} = 0 \text{ if } r(j) > 0 \\ \text{At } x = B\text{:} & \sum_i^n a_i\Gamma_{ik}e^{\lambda_i B} = \pi_k \text{ if } r(k) < 0 \end{array}$$

The n linearly independent equations above define all a_i parameters. The numerical problem arises in the second equation. For the a_i parameters, where $\lambda_i > 0$, $e^{\lambda_i B}$ rises to infinity, as B rises. On the other hand, if $\lambda_i < 0$ the $e^{\lambda_i B}$ parameter converges to zero. This extremity, which depends on the sign of λ_i, makes the equation ill-conditioned, resulting in instability. In the following this paper will present two methods solving this problem.

Chapter 3

The additive decomposition method

In [1], [6] a method is proposed which avoids the numerical-instability of the problem. This method separates the eigenvectors using their sign. It is also applicable when the generator and the drift matrices are level-dependent.

Instead of the transition probabilities this method uses the density function, as the boundary conditions are easier to derive for this case. By differentiating equation (2.1) with respect to x one finds, that the differential-equations determining $f(x)$ take the same form as for $F(x)$. In order to solve the equation one needs to obtain the boundary conditions of the problem.

3.1 Boundary conditions

3.1.1 Boundary conditions at $x = 0$

To solve the differential equation, the boundary values of the density function are needed. First let us consider the $x = 0$ level and a state i with positive drift. To obtain the probability-density function, one needs to calculate the probability, that the fluid is in state $i \in S$ and that the fluid is at an $x \leq \Delta$ level. If Δ is very small, this probability is equal to the sum of the probability that the buffer is empty and $f(t, 0, i)\Delta$. From this we will be able to obtain the density function at level 0 by taking the $\Delta \to 0$ limit.

At time $t + dt$ the $x \leq \Delta$ state can be reached from two states. (It is important note, that dt and Δ are independent of each other.) These states are:

(1) At time t the fluid was in a state $j \neq i$, and in the dt time-interval the state switched to i. It can be proved, that we can assume that the fluid level was varying with r_j during the whole dt time interval. As a result at $t - dt$ the fluid-level had to be at $x \leq \Delta - r_j * dt$. The probability of switching to i in the interval is $Q_{ji}dt + o(dt)$. Consequently the probability of the whole event is:

$$P(Z(t) = j, X(t) \leq \Delta - r_j * dt)Q_{ji}dt + o(dt)$$

(2) At time t the fluid was already in state i, and the environmental process did not switch in the dt interval. Hence the fluid level had to be $x \leq \Delta - r_i * dt$. The probability, that during the dt time-interval no state switches occurred is $(1+Q_{ii}dt)+ o(dt)$:

$$P(Z(t) = i, X(t) \leq -r_i dt)(1 + Q_{ii}dt) + o(dt)$$

By summarizing the two possibilities and using the notation $P(i, X \leq \Delta, t) = P(Z(t) = i, X(t) \leq \Delta)$ one obtains:

$$P(i, X \leq \Delta, t+dt) = \sum_{j=1 \quad j \neq i}^{n} P(j, X \leq \Delta - r_j * dt, t) Q_{ji} dt + P(i, X \leq \Delta - r_i dt, t)(1 + Q_{ii} dt)$$

$$P(i, X \leq \Delta, t + dt) - P(i, X \leq \Delta - r_i dt, t) = \sum_{j=1}^{n} P(j, X(t) \leq \Delta - r_j * dt, t) Q_{ji} dt \quad (3.1)$$

Dividing equation (3.1) by dt and taking the $dt \to 0$ limit one finds:

$$\frac{\partial P(i, x \leq \Delta, t)}{\partial t} + r_i * \frac{\partial P(i, x \leq \Delta, t)}{\partial x} = \sum_{j}^{n} P(j, x \leq \Delta, t) * Q_{ji}$$

The partial derivative with respect to x is the probability density function. The partial derivative with respect to t is equal to zero in the stationary ($t \to \infty$) state. Taking the $\Delta \to 0$ limit the equation takes the form:

$$r_i f_i(i, 0) = \sum_{j=1}^{n} P(j, 0) * Q_{ji}$$

For the states, where $r_j > 0$ $P(j, 0) = 0$ as the fluid immediately drifts away from the zero level. This statement is equal to the boundary conditions in (2.1).

As the states with negative drifts have finite probabilities at level 0, the probability density function for these states does not exist. Therefore one can not use the deduction used before are not valid for them. Nevertheless one can calculate the $f(x \to 0)$ limits. The method used to obtain the equation for these states is similar to the one proposed before:

$$P(i, 0, t + dt) = P(i, x \leq -r_i dt, t)(1 + Q_{ii} dt) + \sum_{j \leq i} P(j, X \leq -r_j dt, t) * Q_{ji} dt \qquad (3.2)$$

Taking the first order Taylor polynomial of $F(i, -r_i dt, t) = P(i, X \leq -r_i dt, t)$ one obtains:

$$P(i, x \leq -r_i dt, t) = P(i, 0, t) - r_i dt * f(i, 0, t) + o(dt)$$

Substituting this to equation (3.2) it takes the following form for stationary states:

$$f(i, 0) r_i = \sum_{j}^{n} P(j, 0) Q_{ji}$$

In conclusion the matrix equation describing the boundary conditions at the $x = 0$ level:

$$f(0) * R = P(0) * Q$$

Where $f(0) = [f(1, 0), f(2, 0), ...]$ and $P(0) = [P(1, 0), P(2, 0), ...]$.

3.1.2 Boundary conditions at the buffer limit

The same method used to calculate the boundary conditions at $x = 0$ can be used at the $x = B$ buffer limit. The result is:

$$-f(B) * R = P(B) * Q$$

3.1.3 Boundary conditions at the threshold level

Let us take a fluid model with a threshold at level a. Both the drift matrix and the generator matrix may change at this level. It is also possible, that for some states the sign of the drift also changes. For example let us consider a state where the drift is positive below the threshold level, and is negative above it. In this case if the fluid reaches this level at this state it stays there. (As the level decreases above this height, and it increases below it.) As a result this state will have a non-zero probability at level a. As a result, the probability-distribution function will be non-continuous, the density function will not exist here. The possible drift changes at the threshold level are shown in Figure 3.1.

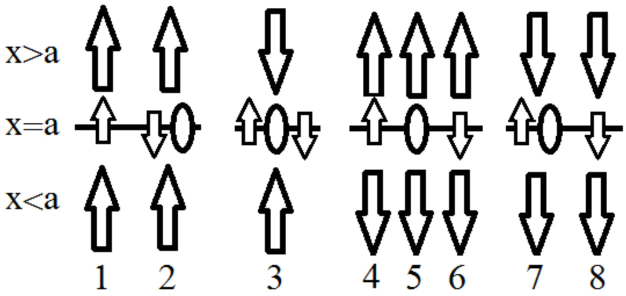

Figure 3.1: The possible behaviours around threshold level. The arrows indicate the sign of the fluid rate. The small arrows, and 0 indicate the fluid rate at the threshold level, which can be zero.

The boundary conditions can be derived with the same technique used in section ??. A detailed analysis of the problem can be found in [5]. The solutions, using the enumeration in Figure 3.1, are summarized below:

1. $p_j(a) = 0$
   $f_j(a+0)r_j(a+0) = f_j(a-0)r_j(a-0) + \sum_{k\neq j} p_k(a)Q_{kj}(a)$

2. $f_j(a+0) = 0$
   $p_j(a)Q_{jj}(a) + f_j(a-0)r_j(a-0) + \sum_{k\neq j} p_k(a)Q_{kj}(a) = 0$

3. $p_j(a)Q_{jj}(a) - f_j(a+0)r_j(a+0) + f_j(a-0)r_j(a-0) + \sum_{k\neq j} p_k(a)Q_{kj}(a) = 0$

4. $p_j(a) = 0$ and $f_j(a-0) = 0$
   $f_j(a+0)r_j(a+0) = \sum_{k\neq j} p_k(a)Q_{kj}(a)$

5. $f_j(a-0)=0$ and $f_j(a+0)=0$
   $p_j(a)Q_{jj}(a)+\sum_{k\neq j} p_k(a)Q_{kj}(a)=0$

6. $p_j(a)=0$ and $f_j(a+0)=0$
   $-f_j(a-0)r_j(a-0)=\sum_{k\neq j} p_k(a)Q_{kj}(a)$

7. $f_j(a-0)=0$ $p_j(a)Q_{jj}(a)-f_j(a+0)r_j(a+0)+\sum_{k\neq j} p_k(a)Q_{kj}(a)=0$

8. $p_j(a)=0$ $-f_j(a-0)r_j(a-0)=-f_j(a+0)r_j(a+0)+\sum_{k\neq j} p_k(a)Q_{kj}(a)$

3.2 Application of additive decomposition

Let us first discuss a fluid model with no thresholds. The instability of the generalized eigenvectors appeared because of the different signs of the generalized eigenvalues. Thus separating them based on their signs is a logical approach to the problem. First let us define a new matrix, A:

$$A \equiv Q * R^{-1}$$

For this definition we needed to assume, that the R matrix is invertible, hence none of the rates for the state space is 0. Often this assumption can not be made. In these cases one can censor out the zero-rate states and a different R and Q matrix can be defined. In [1] it is shown, that by solving the censored fluid model one can obtain the solution for the original model.

The reason for defining A is that it has the same eigenvalues and eigenvectors as the Q, R system. By multiplying equation (2.3) with R^{-1}:

$$\lambda\Gamma RR^{-1} = \Gamma QR^{-1}$$
$$\lambda\Gamma = \Gamma A$$

Using generalized Schur decomposition one can find a matrix T such that:

$$T^{-1}AT = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \lambda A_- & 0 \\ 0 & 0 & \lambda A_+ \end{pmatrix} \tag{3.3}$$

Where the real part of the eigenvalues of A_- is negative, while for A_+ it is positive. Let us denote the number of columns in A_- with n and the number of columns in A_+ with p. (As the matrices are square, n and p also equal with the number of rows.) The algorithm of finding T is the following:

(1) First we use the Schur decomposition create an orthogonal U matrix to separate the eigenvalues:

$$Z = U^TAU = \begin{pmatrix} 0 & Z_{01} & Z_{02} \\ 0 & Z_{11} & Z_1 \\ 0 & 0 & Z_{22} \end{pmatrix}$$

Using the ordered Schur form we can set U so that the eigenvalues of Z_{11} are negative while for Z_{22} they are positive. (Z_{11} and Z_{22} are a square matrices with n and p columns in respect. Z_{01} is a row vector with n columns while Z_{02} has p coloumns..)

(2) Next we eliminate the Z_0 and the Z_11 matrices. First we try to eliminate Z_0. Let us create an V vector with the same size as Z_0 which fulfills the following equation (I represents the identity matrices):

$$\begin{pmatrix} I & V \\ 0 & I \end{pmatrix} \begin{pmatrix} 0 & Z_{01} & Z_{02} \\ 0 & Z_{11} & Z_1 \\ 0 & 0 & Z_{22} \end{pmatrix} \begin{pmatrix} I & -V \\ 0 & I \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & Z_{11} & Z_1 \\ 0 & 0 & Z_{22} \end{pmatrix} \tag{3.4}$$

This results in a Sylvester matrix-equation for V. To elminate Z_1 we define W, an $n * p$ size matrix with satisfies an equation similar to (3.4):

$$\begin{pmatrix} I & 0 & 0 \\ 0 & I & W \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & Z_{11} & Z_1 \\ 0 & 0 & Z_{22} \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & I & -W \\ 0 & 0 & I \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & Z_{11} & 0 \\ 0 & 0 & Z_{22} \end{pmatrix} \tag{3.5}$$

We can obtain T by multiplying the the solutions for (3.4) and (3.4):

$$T = \begin{pmatrix} I & 0 & 0 \\ 0 & I & -W \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} I & -V \\ 0 & I \end{pmatrix} U$$

In order to obtain a numerically stable method one needs to define a new row vector using the T matrix and the probability density function. We will solve the equations for this vector and will use it to find the probability density function:

$$f(x)T \equiv z(x) = [t(x) \qquad u(x) \qquad v(x)]$$

$t(x)$ is a scalar, $u(x)$ is a $1 \times n$ row vector and $v(x)$ is a $1 \times p$ row vector. By multiplying with the inverse of U one finds

$$f(x) = t(x)L_1 + u(x)L_2 + v(x)L_3, \text{ where } T^{-1} = \begin{pmatrix} L_1 \\ L_2 \\ L_3 \end{pmatrix} \tag{3.6}$$

Substituting f defined above to the differential equation, the following equations can be derived:

$$\frac{dt(x)}{dx} = 0$$

$$\frac{du(x)}{dx} = u(x)A_-$$

$$\frac{dv(x)}{dx}R_{22} = v(x)A_+$$

The solutions of the differential equations are:
$t(x) = c$ is a constant
$u(x) = u(0)e^{A_{-}x}$
$v(x) = v(0)e^{A_{+}x} = v(B)e^{-A_{+}(B-x)}$
The second part of the last formula is the key to the stability of the method, because A_{+} has eigenvalues with positive real parts, consequently the eigenvalues of $-A_{+}$ are negative. By using $v(B)$ instead of $v(0)$ the formula for $v(x)$ becomes numerically stable, even for the large values for x. Substituting the solutions in equation (3.6) one obtains:

$$f(x) = cL_1 + u(0)e^{A_{-}x}L_2 + v(B)e^{-A_{+}(B-x)}L_3$$

For fluid models with thresholds at levels x_1, x_2 etc. one needs to decompose to calculate and decompose the A matrices in every homogeneous part. This will result in different T matrices for every part. As a consequence the A_{-},A_{+} matrices and the L_1, L_2, L_3 will be level-dependent. To retrieve a numerically stable formula for $f(x)$ will also change slightly:

$$f(x) = c^iL_1^i + u(x_i)^ie^{A_{-}^i(x-x_i)}L_2^i + v(x_{i+1})e^{-A_{+}^i(x_{i+1}-x)}L_3^i, \text{ if } x_i < x < x_{i+1} \tag{3.7}$$

Using the formula in equation (8), a numerically stable solution can be obtained for the boundary conditions. For example one can always build a homogeneous linear equation system (C):

$$sol * C = 0$$

, where sol is a row vector which contains the c_i, $u(x_i)$, $v(x_{i+1})$ vectors for every i and also the threshold probabilities. C is a well-conditioned matrix consisting of matrices such as $e^{-A_{+}^i(x_i-x_{i-1})}L_3^i$. The solutions of the linear equation is indefinite, as it can be multiplied by a constant, and it will still solve the equation. This indetermination can be removed with the normalization condition:

$$\int_0^B f(x)dx + \sum_{thresholds} p_i = 1 \tag{3.8}$$

Chapter 4

The matrix analytic solution

In [3],[4] another way of solving the numerical instability is presented. It is based on the analogy of the fluid model with QBD processes. The long, detailed derivation of the method will not be discussed here.

It is still assumed, that the R and Q matrices are constant between the threshold levels. In [5] and [6] the author also analyzes a fluid model, where at the threshold levels the R and Q matrices are right continuous. (Which means, that Q, and R are constant between $[x_i, x_{i+1})$, where x_i denotes the threshold level.) The method can be easily generalized for fluid models, where the matrices are not right continuous at the threshold levels. In order to calculate the probability density function a few new definitions and notations are needed:

φ_-^i, φ_+^i

φ_-^i, φ_+^i notates the different states of the S state space. φ_-^i consists of the states where the drift is positive in the $[x_i, x_{i+1}]$ interval. In φ_-^i the drift is negative in $[x_i, x_{i+1}]$. The state space S is the sum of these sets: $S = \varphi_-^i \cup \varphi_+^i$.

φ_u^i, φ_s^i, φ_r^i, φ_d^i

These notations also describe sets of the state space S. $\varphi_u^i = \varphi_+^{i-1} \cap \varphi_+^i$ is the set of the states where the fluid rate is positive both below, and above the i-th threshold level. The other definitions are: $\varphi_s^i = \varphi_+^{i-1} \cap \varphi_-^i$, $\varphi_r^i = \varphi_-^{i-1} \cap \varphi_+^i$, $\varphi_d^i = \varphi_-^{i-1} \cap \varphi_-^i$. The indexes $_u$, $_s$, $_r$, $_d$ will notate the φ_u^i (up), φ_s^i (sticky), φ_r^i (repulsive), φ_d^i (down) sets. For example $Q_{sd}^{(i)}$ is the generator matrix from φ_s^i to φ_d^i. For $x \in [x_0 = 0, x_1]$ there are only two type of sets φ_s^i and φ_u^i. Similarly for $x \in [x_{n-1}, B]$: $S = \varphi_d^n \cup \varphi_s^n$.

$\Psi^{(i)}$, $K^{(i)}$, $U^{(i)}$

$\Psi_{jk}^{(i)}$ is the probability, that starting from $(X = x_i, Z = j)$ the fluid returns to level x_i in a finite amount of time, and that it returns in state k. $j \in \varphi_+^i$ and $k \in \varphi_-^i$. $K^{(i)}$ and $U^{(i)}$ are matrices defined as: $K^{(i)} = (R_+^{(i)})^{-1} Q_{++}^{(i)} + \Psi^{(i)} (R_-^{(i)})^{-1} Q_{-+}^{(i)}$, $U^{(i)} = Q_{--}^{(i)} + Q_{-+}^{(i)} \Psi^i$.

$\Lambda_{++}^{(i)}$, $\Upsilon_{+-}^{(i)}$

$(\Lambda_{++}^{(i)})_{jk}$ is the probability, that starting from $(x_i, j \in \varphi_+^i)$ the fluid level reaches (x_{i+1}) in state $k \in \varphi_+^i)$ before returning to x_i. $(\Upsilon_{+-}^{(i)})_{jk}$ defines the probability, that starting from $(x_i, j \in \varphi_+^i)$ the fluid level returns to (x_i) in state $k \in \varphi_+^i)$ before reaching x_{i+1}.

$\widehat{K}^{(i)}$, $\widehat{U}^{(i)}$, $\widehat{\Lambda}^{i}_{--}$ etc. These matrices have the same meaning as the original ones without the $\widehat{\ }$, but for the reversed model. In the reversed model the drift matrix R is replaced with $\widehat{R} = R$.

4.1 Probabilities at $x = 0$, $x = B$ and at the threshold levels

Let us assume that there are n thresholds $x_1, x_2, ...$, and let us denote the bottom of the container with $x_0 = 0$, and the buffer limit with $x_n = B$. Using stochastic thought the Ω matrix can be derived to calculate the probability values at $x_0, x_1, ...x_n$

$$\Omega = \begin{pmatrix} S^{(0)} & U^{(0)} & 0 & 0 & 0 & 0 \\ L^{(1)} & S^{(1)} & U^{(1)} & 0 & 0 & 0 \\ 0 & L^{(2)} & S^{(2)} & U^{(2)} & 0 & 0 \\ 0 & 0 & ... & ... & ... & 0 \\ 0 & 0 & 0 & L^{(n-1)} & S^{(n-1)} & U^{(n-1)} \\ 0 & 0 & 0 & 0 & L^{(n)} & S^{(n)} \end{pmatrix}$$

Where the following notations were applied:

$$S^{(i)} = \begin{pmatrix} 0 & 0 & \Upsilon^{(i)}_{us} & \Upsilon^{(i)}_{ud} \\ 0 & 0 & \Upsilon^{(i)}_{rs} & \Upsilon^{(i)}_{rd} \\ P^{(i)}_{su} & P^{(i)}_{sr} & P^{(i)}_{ss} & P^{(i)}_{sd} \\ \widehat{\Upsilon}^{(i-1)}_{du} & 0 & \widehat{\Upsilon}^{(0)}_{ds} & 0 \end{pmatrix} \quad U^{(i)} = \begin{pmatrix} \Lambda^{(i)}_{uu} & 0 & \Lambda^{(i)}_{us} & 0 \\ \Lambda^{(i)}_{ru} & 0 & \Lambda^{(i)}_{rs} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad L^{(i)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \widehat{\Lambda}^{(i-1)}_{ds} & \widehat{\Lambda}^{(i-1)}_{dd} \end{pmatrix}$$

$$S^{(0)} = \begin{pmatrix} 0 & \Upsilon^{(0)}_{us} \\ P^{(0)}_{su} & P^{(0)}_{ss} \end{pmatrix} \quad U^{(0)} = \begin{pmatrix} \Lambda^{(0)}_{uu} & 0 & \Lambda^{(0)}_{us} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad L^{(1)} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \widehat{\Lambda}^{(0)}_{ds} \end{pmatrix}$$

$$S^{(n)} = \begin{pmatrix} P^{(n)}_{ss} & P^{(n)}_{sd} \\ \widehat{\Upsilon}^{(n-1)}_{ds} & 0 \end{pmatrix} \quad U^{(n-1)} = \begin{pmatrix} \Lambda^{(n-1)}_{us} & 0 \\ \Lambda^{(n-1)}_{rs} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad L^{(n)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \widehat{\Lambda}^{(n-1)}_{ds} & \widehat{\Lambda}^{(n-1)}_{dd} \end{pmatrix}$$

We assumed, that $Q^n = Q^{n-1}$. $P^{(i)} = 1 + (\Phi^{(i)})^{(-1)} Q^{(i)}$ where $\Phi^{(i)}$ is the matrix of the diagonal elements of $-Q^{(i)}$. $P^{(i)}_i j$ is the conditional probability, that for the next transition in the environmental process, it switches from state i to state j.

Ω can be interpreted as the transition matrix of a Markov chain between the threshold levels. $S^{(i)}$ represents the probability, that the fluid level returns to the i-th threshold level, before reaching any other of the threshold levels, if the walk started from x_i. $U^{(i)}$ is the conditional probability that the fluid level reaches $xi + 1$ before returning to x_i or reaching x_{i-1}. Similarly $L^{(i)}$ is the conditional property of reaching x_{i-1}.

From this interpretation one can understand the different elements of $S^{(i)}$, $U^{(i)}$ and $L^{(i)}$. The lines of these block-matrices represent the initial subsets: The first line indicates the up states (φ_u^i). The other three lines belong to φ_r^i, φ_s^i, φ_d^i in respect. The different elements of the first row of $S^{(i)}$ are the following: The first is the conditional property that the fluid returns to x_i in a state of φ_u^i if it started from the same subset. This is equal to 0, because the initial had positive rate, hence the fluid level ascended immediately above x_i, and it can only return in a state with negative drift. As a result the second member of the line is also 0. The third and fourth are equal to $\Upsilon_{us}^{(i)}$ and $\Upsilon_{ud}^{(i)}$ by definition. The same train of thought can be used to understand the $U^{(i)}$, $L^{(i)}$ matrices and the other elements of $S^{(i)}$.

As Ω is the transition matrix for the threshold levels, it's stationary solution gives us the probabilities at these levels. As the only states with non-zero probabilities are the sticky states of the different threshold levels, these are the parts of the solutions we are interested in. As a result it is logical to filter out the up, repulsive and down states. This can be achieved by switching a few rows and columns in Ω matrix, to attain the form below:

$$\Omega' = \left(\begin{array}{cccc|c} P_{ss}^{(0)} & 0 & ... & 0 & \\ 0 & P_{ss}^{(1)} & ... & 0 & E \\ ... & ... & ... & ... & \\ 0 & 0 & ... & P_{ss}^{(n)} & \\ \hline & F & & & G \end{array}\right)$$

E, F,G are only notations for the blocks in the mixed Ω matrix. Let us define:

$$\Omega'' = \begin{pmatrix} P_{ss}^{(0)} & 0 & ... & 0 \\ 0 & P_{ss}^{(1)} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & P_{ss}^{(n)} \end{pmatrix} + E(I-G)^{-1}F \tag{4.1}$$

It can be proved, that this is the transition matrix for the sticky states. As a result from Ω'' the stationary probabilities for $i \in \phi_s^i$ can be obtained by solving:

$$p\Phi(\Omega'' - 1) = 0 \text{ where } p = (p_s^{(0)}, p_s^{(1)}, ..., p_s^{(n)})$$

$p_s^{(i)}$ is a row vector which consists of the probabilities for the states $i \in \phi_s^i$ at x_i. As p is a solution to a homogeneous linear equation λp will also solve equation (4.1). This indefiniteness of p can be solved by the normalization condition (3.8).

4.2 Calculation of the probability density

From the solution of (4.1) the values of the probability density function can be attained near the threshold levels. The the absolute value of $R^{(i)}$ will be denoted with $C^{(i)}$: $C^{(i)} = diag(|r_j| where j \in S)$.
For $2 \leq i \leq n-1$:
$f_u(x_i{+}0) = (p_s^{(i)} Q_{su}^{(i)} + p_s^{(i-1)} Q_{sr}^{(i-1)} \Lambda_{ru}^{(i-1)} + f_u(x_{i-1}{+}0) C_u^{(i-1)} \Lambda_{uu}^{(i-1)} + f_d(c_i{-}0) C_d^{(i-1)} \widehat{\Upsilon}_{du}^{(i-1)})(C_u^{(i)})^{-1}$

For $1 \leq i \leq n-2$:
$f_d(x_i - 0) = (p_s^{(i)} Q_{sd}^{(i)} + p_s^{(i)} Q_{sr}^{(i)} \Upsilon_{rd}^{(i)} + f_u(x_i + 0) C_u^{(i)} \Upsilon_{ud}^{(i)} + f_d(c_{i+1} - 0) C_d^{(i)} \widehat{\Lambda}_{dd}^{(i)})(C_d^{(i-1)})^{-1}$

Furthermore:
$f_u(x_1 + 0) = (p_s^1 Q_{su}^{(1)} + p_s^{(0)} Q_{su}^{(0)} \Lambda_{uu}^{(0)} + f_d(c_1 - 0) C_d^{(0)} \widehat{\Upsilon}_{du}^{(0)})(C_u^{(1)})^{-1}$
$f_d(x_{n-1} - 0) = (p_s^{(n-1)} Q_{sd}^{(n-1)} + p_s^{(n)} Q_{sd}^{(n-1)} \widehat{\Lambda}_{dd}^{(n-1)} + f_u(x_{n-1} + 0) C_u^{(n-1)} \Upsilon_{ud}^{(n-1)})(C_d^{(n-2)})^{-1}$

The meaning of these formulas can be understood by using the equations for the boundary conditions derived in section 3.1. The first equation can be derived from the formula for the boundary conditions at threshold levels.

$$f(x_i + 0)R^{(i)} - f(x_i - 0)R^{(i-1)} = p^{(i)} * Q$$

The formula is the same as in section 3.1, but it is written slightly differently. For the *up* states of level x_i the equation will be the following:

$$f_u(x_i + 0)R_u^{(i)} - f_u(x_i - 0)R_u^{(i-1)} = p_s^{(i)} * Q_{su}$$

As only the probabilities for the *sticky* states have nonzero value at the threshold levels. $f_u(x_i - 0)R_u^{(i-1)}$ can be expressed with the values of the probability density function at different states and threshold levels, if we condition on the last visits on the threshold levels. By conditioning we mean, that we check the last threshold level the fluid visited and also the state it visited it in. Then we try to find the probability that it reaches x_i in a state of φ_u^i before visiting any other threshold level or state of x_i. This means the analysis of a censored process, similar to the one performed for Ω. The different possible cases for this event are the following:

1. (1)The state can be reached from an *up* state of x_{i-1} if the fluid reaches the x_i level before returning to the starting fluid height, and does so in an *up*state of x_i. The probability of this whole process is $f_u(x_{i-1} + 0)\Lambda_{uu}^{(i-1)}$

2. (2)If the last fluid level was at x_i it may still reach x_i in an state of φ_u^i, if it started in a *down* state, and was able to return before reaching x_{i-1}. By the definition of $\widehat{\Upsilon}$ The probability of this event is $f_d(x_i - 0)\widehat{\Upsilon}_{du}^{(i-1)}$

3. (3)Moreover if the last threshold visited was x_{i-1}, and it was in a *sticky* state, again, it could reach x_i in an *up* state. This requires that at x_{i-1} the environmental process switched to a *repulsive* state: $p_s^{(i-1)} * Q_{sr}^{(i-1)} \Lambda_{ru}^{(i-1)}$

Those are the only three events which contribute to $f_u(x_i - 0)$. The other options are can easily be filtered out. For example φ_u^i can not be reached from a sticky state of x_i, because if it switches to an *up* or a *repulsive* state, the fluid level will rise above x_i immediately. Thus if it will return to x_i it will happen in a state with negative rate at that threshold level. This means that it will return in either a *down* or a *sticky* state. Also, if the environmental process will switch to a *down* state from φ_s^i, it will not add to the probability, because this event is present in the value of $f_d(x_i-0)$. The final equation for $f_u(x_i-0)$ is the following:

$$f_u(x_i-0)C_u^{(i-1)} = f_u(x_{i-1}+0)C_u^{(i-1)}\Lambda_{uu}^{(i-1)} + f_d(x_i-0)C_d^{(i-1)}\widehat{\Upsilon}_{du}^{(i-1)} + p_s^{(i-1)} * Q_{sr}^{(i-1)}\Lambda_{ru}^{(i-1)} \tag{4.2}$$

The origin for the appearance of the C matrices requires a longer derivation, which will not be discussed. For the calculation of the other equations for the boundary values of the probability density function requires a similar algorithm.

By solving these equations, the probability density functions can be calculated.
For $x_i < x < x_{i+1}$, where $1 \leq i \leq n-2$:
$f(x) = (f_u(x_i+0)C_u^{(i)}N_u^{(i)}(0; x-x_i) + p_s^{(i)}Q_{sr}^{(i)}N_r^{(i)}(0; x-x_i) + f_d(x_{i+1}-0)C_d^{(i)}$
$* N_d^{(i)}(x_{i+1}-x_i, x-x_i))(C^{(i)})^{-1}$

For $0 < x < x_1$:
$f(x) = (p_s^{(0)}Q_{su}^{(0)}N_u^{(0)}(0; x) + f_d(x_1-0)C_d^{(0)}N_d^{(0)}(x_1, x))(C^{(0)})^{-1}$

For $x_{n-1} < x < x_n$
$f(x) = (f_u(x_{n-1}+0)C_u^{(n-1)}N_u^{(n-1)}(0, x-x_{n-1}) + p_s^{(n-1)}Q_{sr}^{(n-1)}N_r^{(n-1)}(0, x-x_{n-1}))(C^{(n-1)})^{-1}$

The $N(x_i; x) = N(0; x-x_i)$ matrices can be interpreted as the expected value of the number of events of reaching x before arriving to the threshold levels, assuming that the starting fluid height was x_i. The method of deriving of the equations above is the same as it was for the boundary values of $f(x)$. It can be proved, that the N matrices can be calculated with the following formula:

$$N^{(i)} = \begin{pmatrix} N_+^{(i)}(0,x) \\ N_-^{(i)}(x_{i+1}-x_i, x) \end{pmatrix} =$$
$$\begin{pmatrix} 1 & e^{K^{(i)}(x_{i+1}-x_i)}\Psi^{(i)} \\ e^{\widehat{K}^{(i)}(x_{i+1}-x_i)}\widehat{\Psi}^{(i)} & 1 \end{pmatrix}^{-1} \begin{pmatrix} e^{K^{(i)}x} & e^{K^{(i)}x}\Psi^{(i)} \\ e^{\widehat{K}^{(i)}(x_{i+1}-x_i-x)}\widehat{\Psi}^{(i)} & e^{\widehat{K}^{(i)}(x_{i+1}-x_i-x)} \end{pmatrix}$$

4.3 The calculation of Ψ

As seen in section 4.2 the only thing one needs to calculate both the probability density function and the probabilities at the threshold levels is $\Psi^{(i)}$ for every i. In [3] it is shown that it can be computed using equation (4.3).

$$G = A_- + A_0 G + A_+ G^2 \tag{4.3}$$

$$A_+ = \begin{pmatrix} \frac{1}{2} * 1 & 0 \\ 0 & 0 \end{pmatrix} \quad A_0 = \begin{pmatrix} \frac{1}{2}P_{++} & 0 \\ P_{-+} & 0 \end{pmatrix} \quad A_= \begin{pmatrix} 0 & \frac{1}{2}P_{+-} \\ 0 & P_{--} \end{pmatrix}$$

The minimal nonnegative solution of (4.3) will take the form:

$$G = \begin{pmatrix} 0 & \Psi \\ 0 & V \end{pmatrix}$$

The numerical stability of this approach can be understood by it's stochastic interpretation. As mentioned before the $N_+^{(i)}(0; x - x_i)$ matrix equals the expected number of visits of a fluid-level $x_i + x$ after visiting x_i in a state with positive drift, and before returning to level x_i. Consequently $N_+^{(i)}(0, x)$ monotonously decreases with x, as it becomes less probable that the fluid reaches x before returning to the starting threshold level. Mathematically is only possible if $K^{(i)}$ is a stable matrix. Similarly it can be proven that $\widehat{K}^{(i)}$ is also a stable matrix. This means that all linear equations are numerically stable, even for the large values of x.

Chapter 5

Comparing the two methods

I have implemented the two methods summarized above using MATLAB, and compared them based on their numerical stability and speed. Approximately both methods can be divided into three parts. The schematic representations of the methods can be seen in Figure 5.1.

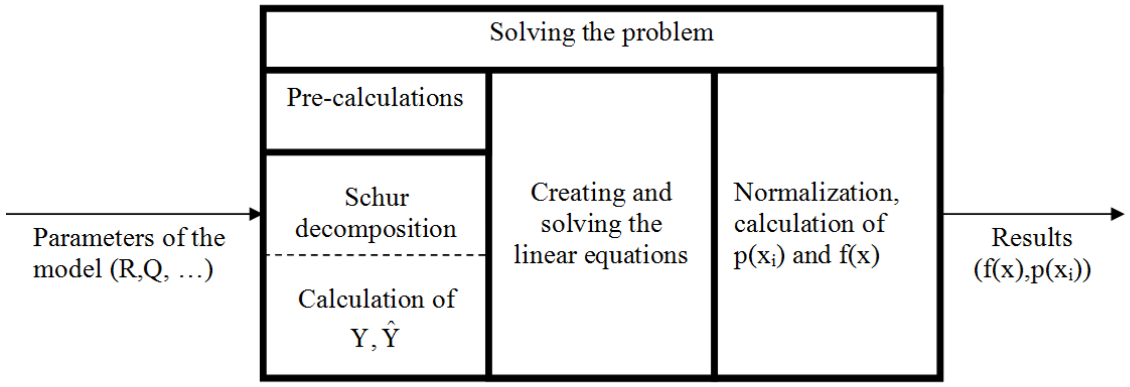
Solving the problem
Pre-calculations
Schur decomposition
Calculation of Y, Ŷ
Creating and solving the linear equations
Normalization, calculation of p(xᵢ) and f(x)
Parameters of the model (R,Q, …)
Results (f(x),p(xᵢ))

Figure 5.1: The different sections of the algorithms

5.1 Numerical stability

Both methods are quite successful in computing the stationary values of the probability density function. The main problem which may occur arises when one tries to decompose the matrix during the additive decomposition method, or when one tries to solve (4.3) in the matrix analytic method. Although it happens rarely, but the problem may arise in an infinite buffer system when the mean drift is close to zero and $x \to \infty$. The mean drift is the expected value of the rate with which the fluid level rises at a certain level. It can be calculated using the stationary distribution π, defined at the beginning of the paper:

$$d = \sum_{i}^{n} \pi_i r_i \tag{5.1}$$

As both Q and R are allowed to be level-dependent, the value of d may also change with the fluid level. (In this case it may change at threshold levels.)

If $\lim_{x\to\infty} d(x) = 0$ means that the fluid has a nonzero probability of rising above any limits. (As the probability of rising and decreasing would be the same at any given point.) This would result in an instability, such systems will not be analysed in this paper.

On the other hand it is possible, that the mean drift is a negative number with a very small absolute value. In this case the stationary probability density function would decrease slowly with the fluid level. Because of this at least one of the generalized eigenvalues in the spectral representation would have to be a negative number with a very small absolute value. Because of the numerical representation of this eigenvalue is not exact, this could result in high inaccuracy. As the additive decomposition method is based on the spectral representation, this may present a limit for it's applicability.

As a consequence the matrix analytic method may also have problems when calculating the properties of the fluid queue. When examining QBD solvers one can usually make the following picture of their algorithm: For large buffers when calculating Υ they only use those paths in the calculation which have a small maximum value of x. A path is the trajectory of the fluid-height. Then as the algorithm progresses they start to consider the paths which reach higher fluid levels. If the mean drift is small, then the algorithms will converge fast, as it is very likely, that the fluid will not reach high levels. For large drifts it is quite possible that the fluid will reach higher levels, but from these heights it is likely that they will not return to the starting level. Therefore for higher drifts the first few paths, with smaller maximum, will be more important, the algorithms will converge fast. The problem will arise for mean drifts close to zero, as all fluid-heights are easily reachable from all levels, the iteration will converge much slower.

To analyze this problem, both methods were implemented for a homogeneous infinite buffer system. The programs were tested for different mean drift values. The different drift values were obtained by changing the values of both Q and R matrices. For the Schur decomposition, S.N. Bangert's Lyapunov equation solver was used. A few algorithms were tested for the quadratic equations of the matrix analytic method. (Newton Iteration, Logarithmic Reduction, Invariant Subspace Iteration, Functional Iteration, Cyclic Reduction) The most promising results, both in speed and stability, were calculated using a cyclic reduction program created by Benny VanHoudt [2].

The probability mass at zero height was calculated with both methods. The results were same for both until a mean drift as small as $d = -10^{-6}$. Then a minor relative difference appeared in their values, around 0.01 which started to increase rapidly, as the absolute value of the mean drift was decreasing. As the d was increasing, (it's absolute value was decreasing,) one would expect that the probability mass at $x = 0$ level would decrease. It appeared that for lower drifts than $d = -10^{-8}$ the additive decomposition method was unable to fulfill this condition, while no limit was found for the matrix analytic method. Therefore it is probable that the matrix analytic method is the more powerful one. It is important to mention that from all the solvers analyzed, only the cyclic reduction was able solve the problem for every mean drift.

5.2 Comparison based on calculation speed

To examine both algorithms, numerous tests with different parameters were conducted. The programs were tested on a Pentium 4, 3 GHz computer, with 512 MB RAM (Physical Address Extension).

5.2.1 Dependence on number of states

The first tests were made in order to define how much time it took for the methods to calculate the probability masses at the threshold levels and the parameters of the probability density function, and how they depended on the cardinality of the state-space. Also we were interested whether the cyclic reduction QBD solver and the Lyapunov equation solver could cope with the large matrices. The same program as in the previous section was used to test this attribute of the algorithms. The largest cardinality was 900 but both methods could solve the problem, the relative differences in the results were less than 10^{-9}. On the other hand there were significant differences in their runtime as it can be seen in Figure 5.2.

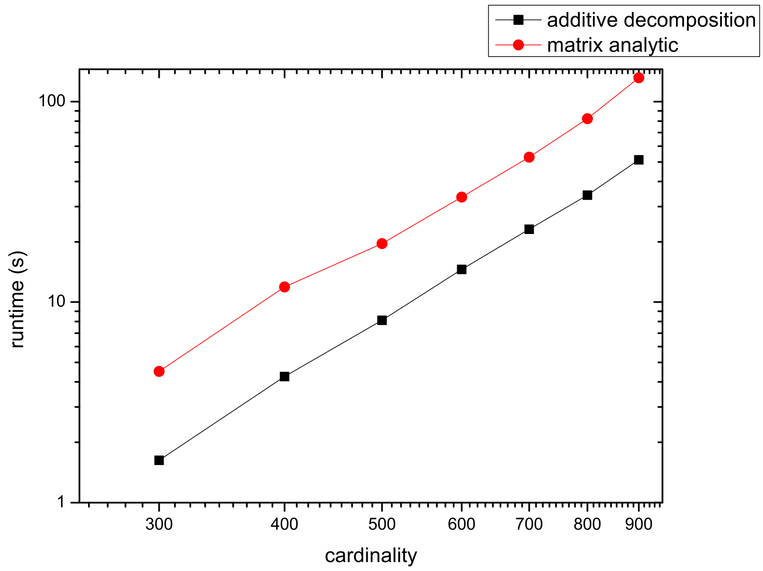

Figure 5.2: The cardinality-dependence of the runtime for the additive decomposition and the matrix analytic method on log-log scale

For both algorithms, the runtime depended on the cube of the cardinality. The fits on the curve were:

Additive decomposition: $t = (6.87 \pm 0.07)10^{-8} * x^3$

Matrix analytic: $t = (15,62 \pm 0.14)10^{-8} * x^3$

Both from the figure and the fits it is visible that the additive decomposition is about 2.31 ± 0.03 times faster than the matrix analytic method. It may be possible to speed up the cyclic reduction with about 30% but that would still leave the additive decomposition method the faster.

5.2.2 Calculation time for the linear equations

For this section we were interested how the methods scale with the number of thresholds. Roughly, the programs can be divided into two sections, one is solving the QBDs for the matrix analytic method or the Schur-decomposition for the additive-decomposition. The second part is the time it takes to solve the linear equations. The runtime of first part can be analyzed with a homogeneous fluid model with infinite buffer system, which is the same type of systems which were used in the previous sections.

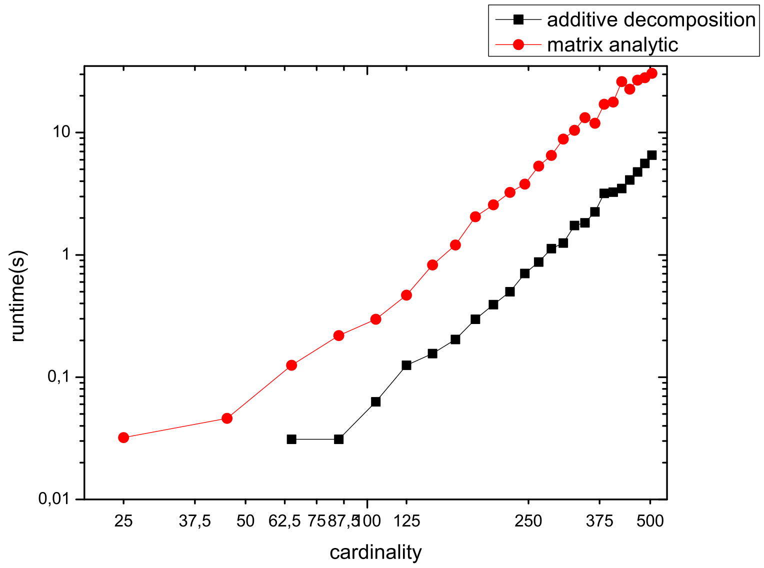

Figure 5.3: The runtime for the QBD and the Schur decomposition on logharitmic scale.

It is visible, that approximately both algorithms scale the same way with the time: $t = \tilde{e}^{(7.82\pm0.45)*10^{-3}*x}$

Also the Schur decomposition proved to be a much faster method: $\frac{t(Schur)}{tQBD} = 0.116 \pm 0.03$

Usually it does not take much time to solve the linear equations, as a result measuring the runtime can get quite inaccurate. For this task a different program was used, in which the number of thresholds can be changed to any number. It was used to measure runtime for different threshold numbers and state space-cardinality. The results are summarized in Figure 5.4.

From the graphs one can see that again, both methods scale the same way with the number of states and the number of thresholds. On the other hand the matrix analytic method gave much faster results. The reason for this is, that while for the additive decomposition method at every threshold there is an equation for every state, for the matrix analytic method Ω'' has the equations only for the sticky states at the threshold levels. This was achieved by filtering out the *up*, *repulsive* and *down* states of the Ω matrix. The time needed for this algorithm was also added to the "calculation time for the linear equations". As about one fourth of the states are sticky at every threshold, and the Gauss elimination is a cubic method, this property itself would make the matrix analytic method much times faster. Also, because in the tests random matrices were used,

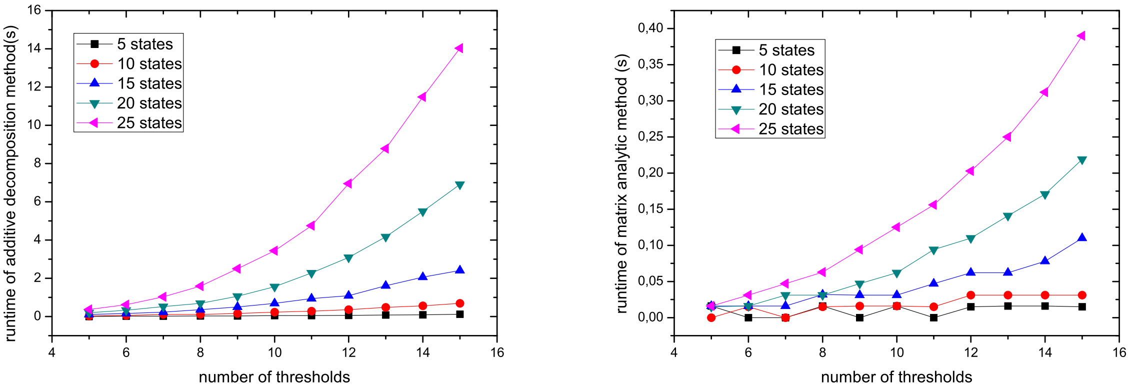

Figure 5.4: The runtime of the additive decompositon and the matrix analytic method for solving the linear equations

the number of sticky states was not always one fourth of all the states, as a result these graphs look a bit more stochastic.

5.2.3 Calculation of the probability density function

Often the most important part of the results is the values of the probability density function at different points. In addition one is usually interested in the values at a numerous points, so the calculation speed is a critical property of the algorithms. To test this characteristic of the methods, the same multi-threshold programs were used as in the previous section. The functions were calculated in ten thousand points. The most resource-consuming part of the calculation is the matrix exponential parts which are present in both methods. The only difference in the two methods are the size of the matrices. While in the matrix analytic approach the sum of the sizes of K and $\widehat{K}$ equals the number of states. For the additive decomposition the sum of the sizes of G_1 and G_2 equals the cardinality of the state space minus one. As a result it is expected that the latter will be the faster one.

The results are coherent with the theoretical expectations. As computing the stationary probability density function is usually one of the main aims of the calculations, this difference in their speed is of great importance.

5.3 Summary

In our comparison the matrix analytic method performed as a robust, reliable approach. For every test, the results were plausible, and no limits were found for it's applicability. On the other hand the additive decomposition algorithm appeared slightly less stable. When examining the runtimes it seemed that none of the methods were faster, their speed compared to each other was highly dependent on the system. An example of the runtimes can be seen in table below. The threshold number was 14 and the cardinality

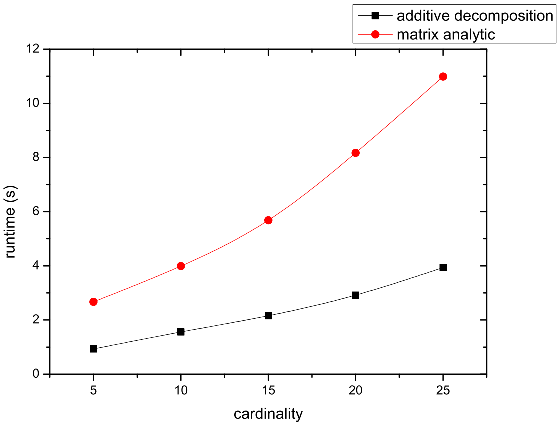

Figure 5.5: The calculation time for the density function

of the state space was 25. The summary of the results can be seen in Figure 5.6.

Runtime	Matrix analytic (s)	Additive decomposition (s)
Cyclic red./Schur dec.	0.034	0.015
Solving the equation	0.312	11.483
Calculating the probability density	2.252	0.817
Sum	2.598	12.315

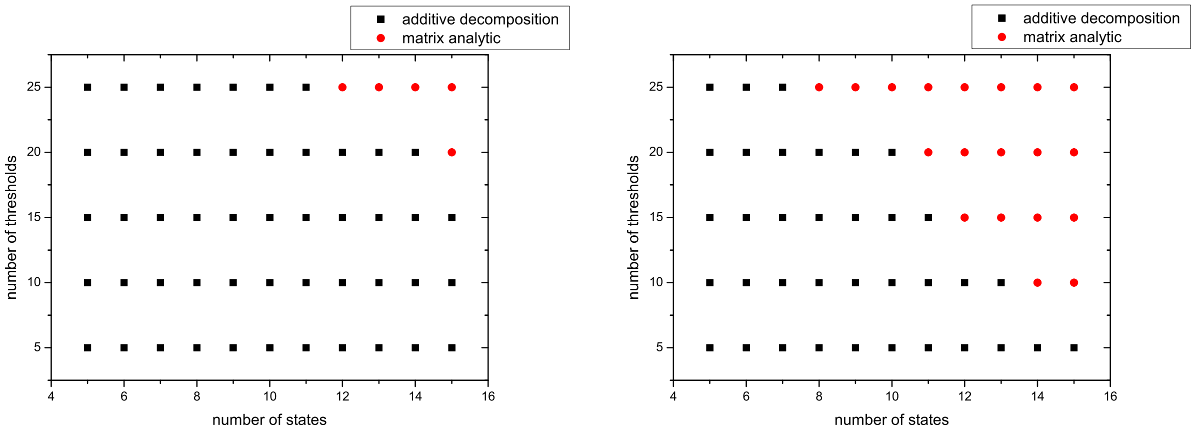

Figure 5.6: These figures show which method was faster if the density function was calculated in 10000 and 2000 points

From Figure 5.6 it is visible that the additive decomposition method is more useful for lower threshold numbers and cardinality, and if one is interested in the density function in a lot of points.

In conclusion, while the matrix analytic method performed more stability, but for most systems the additive decomposition method was also quite stable. The main difference of the methods is that their speed scale differently from the system.

Chapter 6

Application of the fluid models for RED processes

As it is mentioned in the introduction, one of the possible applications of stochastic fluid models is the analysis of servers of telecommunication networks. The fluid level represents the amount of demands in the system, and the arrival and the service process is modulated by an environmental Markov-process defining R and Q. Let us assume that there are N users in the system with identical properties. We presume that either they are in an ON state, when the system receives requests from them continually, or they are in an OFF state when they do not generate requests. As the fluid model needs the environmental process to be memoryless, we assume that the time for which the users stay in an ON or an OFF state are defined by an exponential probability distribution. Finally we assume, that the server has a limited serving speed, and may also reject some requests with $1 - s$ probability. This last functional property of the server is called the "random early detection" (RED) mechanism.

For the ON state we will notate the expected value of the exponential distribution with $1/\alpha$. For the OFF state it will be denoted by $1/\beta$. Furthermore the rate with which the user produces the requests will be denoted by r. The speed of the serving process is c. As mentioned before, the server will only accept the new requests with s probability. A representation of the system can be seen in Figure 6.1 .

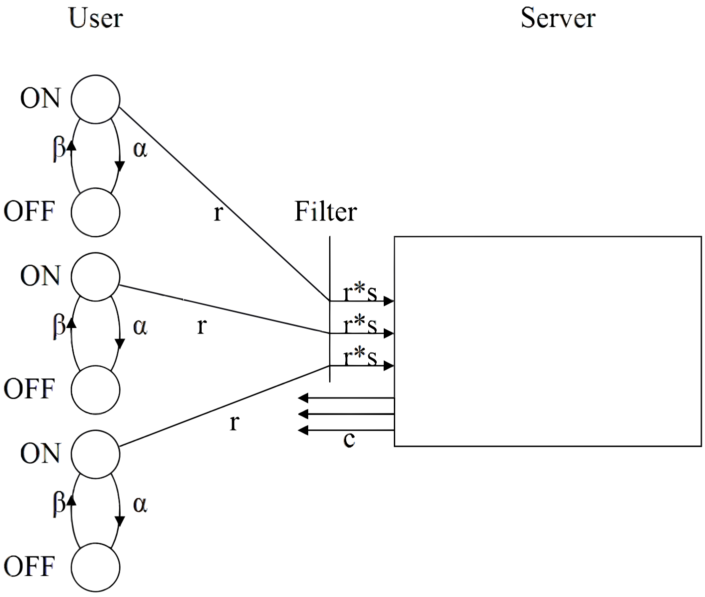

Figure 6.1: The schematic visualization of the process

6.1 The calculation of the Q and R matrices

Let us differentiate the states of the environmental process based on the number of users in ON state. If n users are ON at time t, then in the following a dt interval three things may happen

1. One or more of the $N - n$ users in OFF states may switch to an on state. As the transitions between states are independent for the different users, and the probability distribution for this event is continuous with time, it is highly unlikely that more users will switch their states in a small dt interval. If we want to calculate this probability, we can use the memoryless property of the exponential distribution: If there were no transitions before t, than the probability of a transition at the [t,t+dt] is the same as in [0,dt]. The users which switch to ON will be denoted with i, j will be used for the ones which were in OFF state and stay there, and k for those which stay in ON.
   P(One user switches ON, all others stay in the same state between [t,t+dt]|n were ON at t)=P(One user switches ON, all others stay in the same state between [0,dt]| n were ON at 0)=$\sum_i$*[P(the i-th user switches ON in [0,dt]|it was OFF at t=0)*$\prod_j$*P(the j-th user stays OFF between [0,dt]|it was OFF at t=0)*$\prod_k$*P(the j-th user stays ON between [0,dt]|it was ON at t=0)]=*$\sum_i \beta e^{-\beta 0} dt * \prod_j (1 - \beta e^{-\beta 0} dt) * \prod_k (1 - \alpha e^{-\alpha 0} dt) = \sum_i \beta dt + o(dt) = (N - n)\beta dt + o(dt)$

2. One of the n users in ON state switch OFF, similarly, the probability of this transition is:
   P(One switches OFF in [t,t+dt] | n were ON at t)=$n\alpha dt + o(dt)$

3. All users stay in the same state they were at time t: *P(No transition occurs in [t,t+dt] | n were ON at t)*$=1 - n\alpha dt - (N-n)\beta dt + o(dt)$

If the lines of the Q matrix correspond to the number of states ON, then it will take the following form:

$$Q = \begin{pmatrix} -N\beta & N\beta & 0 & 0 & 0 & 0 \\ \alpha & -\alpha-(N-1)\beta & (N-1)\beta & 0 & 0 & 0 \\ 0 & 2\alpha & -2\alpha-(N-2)\beta & (N-2)\beta & 0 & 0 \\ 0 & 0 & ... & ... & ... & 0 \\ 0 & 0 & 0 & (N-1)\alpha & -(N-1)\alpha-\beta & \beta \\ 0 & 0 & 0 & 0 & N\alpha & -N\alpha \end{pmatrix}$$

The calculation of the R matrix is an easier task. In a state where n users are ON, they produce $n * r$ requests. But the server does not accept all requests, they are filtered out randomly, with $1 - s$ probability. In addition, the serving process is constantly c. Hence in our model the fluid level in the buffer system will only rise with $n * r * s - c$.

$$R = \begin{pmatrix} -c & 0 & 0 & 0 & 0 & \\ 0 & rs-c & 0 & 0 & 0 & 0 \\ 0 & 0 & 2rs-c & 0 & 0 & 0 \\ 0 & 0 & ... & ... & ... & 0 \\ 0 & 0 & 0 & 0 & (N-1)rs-c & \\ 0 & 0 & 0 & 0 & 0 & Nrs-c \end{pmatrix}$$

6.2 Calculation of the loss (L)

The loss is the number of requests that the system is unable to serve. This may be caused by two phenomenons. The first is because of the filtering of the RED processes. In a state, where n users are producing request with r rate, this loss is equal to $L = (1-s)nr$. The second reason for the loss appears if the fluid reaches the buffer-limit of the system. In this case, beside the loss because of the filtering, the server may also loose requests which it is unable to store. As the requests are served with a speed c, this is also the maximum speed the system is able to accept new requests, if it is full. (It can be interpreted as if the model was full with fluid, but we would still like to pour water into it, then some of it would spill out of the storage.) The additional loss at this level is the following: The requests pass the filter, with s probability, but the server is only able to store them with c speed: $L = snr - c$.

The minimal available loss for the system is defined by the speed of the serving process. Although the filtering method may change the probability-density function of the fluid-level, but it is only achieved by loosing some of the requests before they get stored in the server. This does not result in increased serving speed, therefore the loss can not be decreased with RED processes. An approximation of the relative loss (L_{min}) can be calculated using the mean drift, defined in 5.1. If the mean drift, d is smaller then c, then it is unlikely that the fluid will reach the buffer-limit, therefore the loss is close to

0. If $d > c$, and the filtering is turned off, then eventually the fluid level will reach the buffer-limit, therefore some of the requests will "spill out" of the storage.:

$$L_{min} = \frac{mean \quad drift - storing \quad speed}{mean \quad drift} = \frac{d-c}{d} \tag{6.1}$$

This is just an approximation, as even if $d < c$, it is possible, that there are states wher enough users are in the ON state to make the fluid rate positive. If the fluid height reaches the buffer-limit in such a state eventually some fluid will spill out.

As we are able to analyse fluid models, where the R matrix changes with the fluid level, we can use our implementations for systems, where the passing probability changes with the fluid height. (Although we must limit the system, so that s only changes at certain threshold levels.) The exact loss of the system consists of two parts:

1. The system may loose some requests because of the filtering process. If the environmental process is in state i, then at level x the rate with which the requests are lost is $r_i(1 - s(x))$. (r_i is the rate for state i.) Thus the expected value of the loss is:

   $$L_1 = \int_0^B r_i(1 - s(x))f_i(x)dx + \sum_{j,k} p(x_j, k)r_k(1 - s(x_j)$$

   Where f_ix is the stationary probability distribution for state i, and $p(x_j, k)$ is the probability at threshold level x_j for state k. (The 0 level and the buffer limit are also among the x_j-s.)

2. Moreover requests may be lost at the buffer limit, even if they pass the filter, as the system is only able to store them with c speed:

   $$L_2 = \sum_k p(B, k)(s(B)r_k - c)$$

   For the states, where $s(B)r_k > c$.

The relative loss is: $L = \frac{L_1+L_2}{\int_0^B r_i f_i(x)dx + \sum_{j,k} p(x_j,k)r_k}$

6.3 Benefits of RED processes

The advantage of the RED method lies in it's capability to change the stationary probability distribution function. An extreme example for this can be, that if the server does not accept any new requests if the fluid level reaches a certain height, then the fluid can not ascend above this level, the probability distribution will be 0 above this altitude. Because the filtering can only decrease the incoming rate, it will increase the probability that the fluid is in a lower level. Hence the expected value of the fluid level can be lowered with this method. This has a positive effect on the amount of time a user has to wait before being served. This can be understood by a simple model. Let us assume, that the requests are served in the same order they arrive. (These are the "first in, first out" systems.) For our approach this means, that the new requests are added to the top of the "fluid", and their serving time depends on the height of the fluid beneath them. For a new request the expected serving time (T) is:

$$T = \frac{Expected \quad fluid \quad height}{Decreasing \quad speed} = \frac{E(X)}{c} \tag{6.2}$$

A further benefit is that some protocols detect if their requests are rejected, and are able to alter their demands towards the server accordingly. Thus if the server is under high load, and rejects some requests, then it may help to decrease the system's actual demand.

6.4 Applying the method, results

We can use our fluid model for the optimization of the $s(x)$. The object was to minimalize the expected value of the fluid height. It is also important, that we do not loose to many requests in our system. Therefore a restriction was added to the problem: The possible values of $s(x)$ was limited, so that the overall loss, L could not exceed a certain number. In our model AMR voice traffic was analysed. The parameters for this system are the following:

α	$2/3\frac{1}{s}$
β	$1\frac{1}{s}$
r	$12.2kbps$
Number of users	25

The mean drift for this system is $d = 183kbps$. A system with $c = 190kbps$ serving speed and $B = 30kb$ buffer limit was analysed.

With our models we are only able to calculate fluid models which are made of homogeneous sections. (The R,Q parameters of the systems only change at threshold levels.) Hence the $s(x)$ function I have used also had this property. The buffer was divided into three or six parts. I did not try to optimize the threshold levels, they were fixed during the optimization process. Therefore the only parameter that could be changed was the value of $s(x)$ in the different sections. For example if the system was divided into three sections, then $s(x)$ could be defined by three numbers (s_1, s_2 and s_3): it's value in $[x_0, x_1]$, in $[x_1, x_2]$ and in $[x_2, x_3]$. (x_i represents the threshold levels, $x_0 = 0$ is the bottom and $x_3 = B$ is the top of the buffer.) So s is defined by the $[s_1, s_2, s_3]$ vector.

I used the gradient method to optimize the system: We start from a random $[s_1, s_2, s_3]$ vector, and check whether there is a $[s_1+\Delta s_1, s_2+\Delta s_2, s_3+\Delta s_3]$ vector in it's neighborhood which has a smaller expected value for the fluid height. (Δ represents, that Δs_i is small.) If there is one, we check whether the L loss, which belongs to this new s function, is smaller then the maximum acceptable value of the loss, what we defined previously. If L is to high we neglected this vector. From the new s vectors we select the one with the smallest expected fluid level. Then we check the neighborhood of this vector, and continue with our algorithm. We stop the algorithm, when it is unable to find a better

$s(x)$ function then the current one. The matrix analytic method was used to compute the solution of the system, as it seemed more stable when it was compared with the additive decomposition in Chapter 5.

Our results were the following:

	1	2	3
Threshold levels	0,10,20,30	0,5,10,15	0,5,10,15,20,25,30
Limit of loss	0.1	0.1	0.1
Expected fluid height without filtering	11.0633	11.0633	11.0633
Loss without filtering	0.0467	0.0467	0.0467
Results			
Minimal fluid height	2.4931	1.626	1.3839
Loss	0.0999	0.0985	0.0995
Optimal s	s_1=0.9465 s_2=0.8197 s_3=0.7521	s_1=1.0000 s_2=0.8368 s_3=0.9637 s_4=0.7200 s_5=0.8800 s_6=0.9570	s_1=0.9437 s_2=0.8209 s_3=0.7200

When comparing the first and the second column, we see, that for smaller L limit the expected fluid level rises. It is logical, as the only way we can decrease the expected fluid is by loosing some requests. (This is the reason why the optimal system had a loss so close to the limit.) Furthermore, in the parameters of the third and the first column the only difference is in the number of thresholds. By duplicating the number of sections the expected value of the fluid level has decreased by 44%. This is also plausible as we can optimalize a higher number of parameters.

The density functions can be seen in Figure 6.2. From the first graph one can understand why the s function is high for the lower fluid height. It is quite probable that the fluid is at this level, thus a slight decrease in the passing probability could result in a large increase in the loss. This is the part that mainly defines the loss of the system. Hence, the passing probability needs to be high because of the limit for L. For higher levels the probability density function is close to zero, even the small passing probabilities do not affect the system too much.

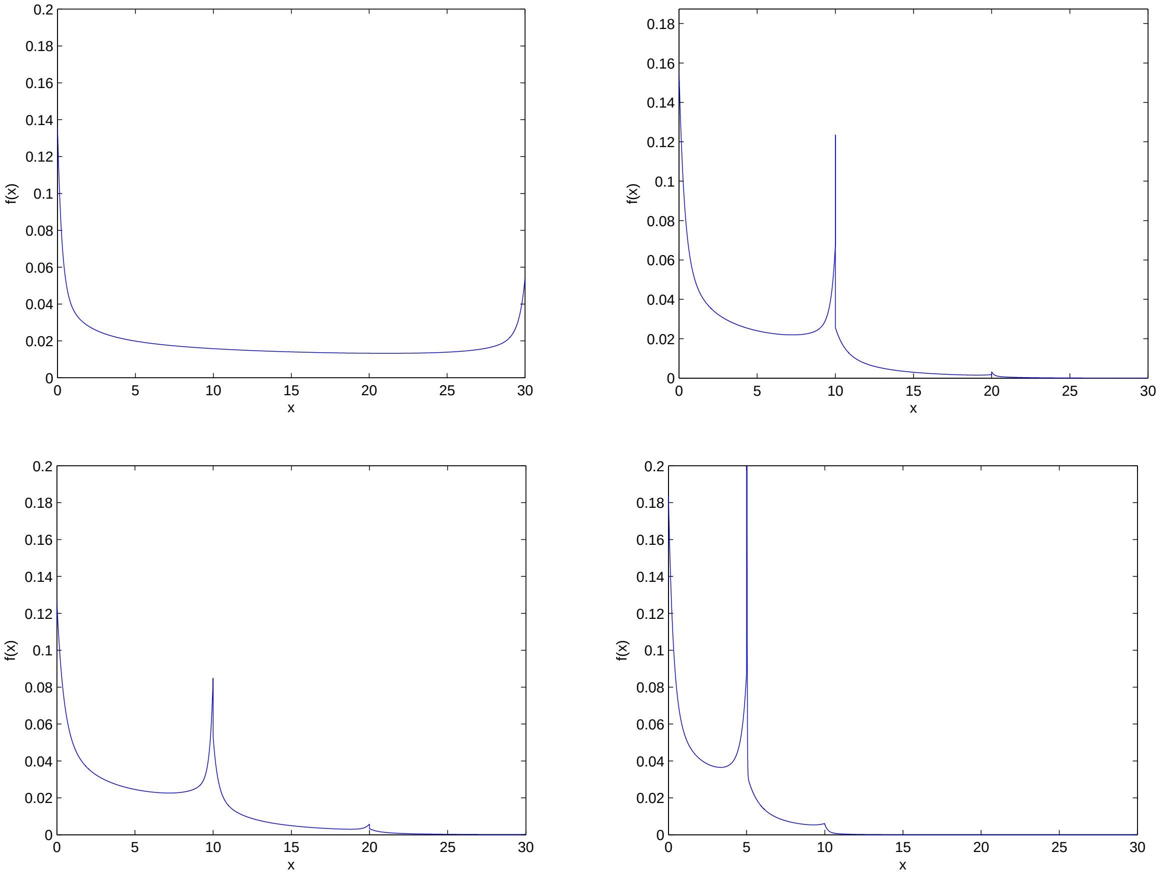

Figure 6.2: From left to right, top to bottom: The first graph shows the density function when the filtering is turned off. The other three correspond to the columns of the table of results.

Chapter 7

Summary

This paper aimed at analysing fluid models. In the first part the basic properties of the system and it's numerical problem for large buffer-limit was discussed. Two methods were presented in Chapter 3 and 4 which solve the problem. When I analyzed these methods, both performed quite well, but the matrix analytic method seemed to be the more robust. The speed of the methods were highly dependent on the problem we tried to solve. For certain systems the matrix analytic method was the faster, but for other problems the additive decomposition had significantly smaller runtime.

In the last chapter I applied the model for a practical problem. We tried to optimize the request-filters of a server to decrease the expected fluid level, while keeping the overall loss as small as possible. The results seemed plausible, the methods probably converged to a local optimum. From the promising results I believe, that this model could be an effective method for a wide range of problems, with stochastic properties.

Bibliography

[1] N. Akar and K. Sohraby. Algorithmic solution of finite markov fluid queues. In *International Teletraffic Congress*, Berlin, Germany, Sep. 2003.

[2] S. Steffé D. A. Bini, B. Meini and B. Van Houdt. *Structured Markov chains solver: software tools.*

[3] Ana da Silva Soares and Guy Latouche. Matrix-analytic methods for fluid queues with finite buffers. *Performance Evaluation*, 63(4-5):295 – 314, 2006.

[4] Ana da Silva Soares and Guy Latouche. Fluid queues with level dependent evolution. *European Journal of Operational Research*, 196(3):1041 – 1048, 2009.

[5] M. Gribaudo and M. Telek. Stationary analysis of fluid level dependent bounded fluid models. *Performance Evaluation*, 65(3-4):241–261, 2008.

[6] H.E. Kankaya and N. Akar. Solving multi-regime feedback fluid queues. *Stochastic Models*, 24(3):425–450, 2008.

[7] V. G. Kulkarni. Fluid models for single buffer systems. In J. H. Dshalalow, editor, *Models and Applications in Science and Engineering*, Frontiers in Queueing, pages 321–338. CRC Press, 1997.