Verified Factorization Methods for SMP/G/1 Queueing Systems and their Interplay in an Integrated Problem-Solving Environment[*]

Sebastian Kempken and Wolfram Luther[†]
Universität Duisburg-Essen, Duisburg, Germany
sebastian.kempken@gmail.com

Abstract

Semi-Markov processes (SMPs) are adequate models for correlated data traffic in service-integrated communication networks. In recent works, we have developed both modeling techniques and verified factorization methods for the analysis of queueing systems of GI/G/1 and SMP/G/1 types. These methods allow a quick and validated computation of workload distributions at the respective node.

In this paper, we study enhancements and alternatives to these known methods. Our objective is to yield more accurate results for small SMP models, such as described previously in literature, as well as to be able to analyse larger SMP/G/1 models, for instance, resulting from modeling video traffic.

These enhancements include a modified polynomial factorization technique as well as an application of the successive over-relaxation (SOR) technique to the verified Wiener-Hopf factorization.

We give a brief overview of how we combine these methods in an integrated environment and the benefit for modeling and analysis of traffic in communication networks. Using this environment, we illustrate the particular techniques on two examples.

Keywords: verified methods, queueing systems, semi-Markov processes
AMS subject classifications: 65G20, 60K25, 90B22, 94A05

1 Introduction

Today, a declared goal of the telecommunication industry is the convergence of previously separate networks, that is, the objective of migrating both circuit-switched voice and packet-switched data networks to a single packed-switched, service-integrating

[*]Submitted: February 2, 2009; Revised: February 9, 2010; Accepted: March 1, 2010.

[†]This work and the software development presented was carried out as part of an ongoing research project funded by the German Research Council *(Deutsche Forschungsgemeinschaft, DFG).*

network. The services of the network are not limited to voice and data, video streaming is gaining an increasing share of today's internet traffic. The integration of all these services, in order to be useful, requires a proper management of the quality of service (QoS). The Internet Engineering Task Force (IETF) has proposed two approaches: Integrated Services [21] and Differentiated Services [3]. The former relies on a reservation of the required resources, and therefore it is necessary to know the requirements in advance. The latter describes a classification and prioritization of particular data packets. The classification has to take place in edge routers, the core routers of a network can handle the packets according to a given prioritization scheme. In either case, proper methods for the modeling of data traffic and methods for the analysis of these models in order to determine the resources required are crucial.

The resource requirements are unpredictable from a number of viewpoints, this includes the amount of transmission time and volume, the number of network resources available, randomly changing workloads, routes and system parameters and failure events. However, the quality of the network services and the experience of the user strongly depends on those properties.

Stochastic models are a good approach to describe the unpredictable circumstances in communication networks. We employ random variables to describe both the arrival process and the available capacity. At least, two ways are feasible: On the one hand, one may consider the interarrival times between events like arrival of packets, flows, connections or other relevant units, this is corresponding to the classical approach regarding queueing and service systems. On the other hand, time-slotted models can be applied that rely on the counting functions in fixed time intervals. The respective analysis, when it comes to SMP/G/1 queueing systems, is equivalent [15, 6, 7]. In brief, a semi-Markov process is a generalization of a Markov process. The holding times for the particular states are given by arbitrary state- or transistion-specific distributions. In the context of communication networks, the particular states may be identified with corresponding data rates on a link.

Since the capacity on each transmission link is limited, discrete time models with finite distributions are appropriate. This includes deterministic service as the simplest case, however, various autocorrelation structures of Internet traffic are taken into account using semi-Markovian modeling.

The question of how to create appropriate models for given data traffic has been discussed for a long time [17, 14, 8, 19]. The greatest difficulty lies in capturing the autocorrelation of the original traffic, which is important for accurate workload predictions. Video traffic, which represents an increasing part of today's Internet traffic, shows a bursty behavior, i.e., a high level of autocorrelation.

Methods that attempt to overcome these difficulties when determining the parameters of a SMP model of an arrival process are described in [11]. Using these methods, original video traffic as well as aggregated traffic from different types of sources is described by SMP models in a small state space, preserving the autocorrelation function of the original data series. To increase the accuracy of these models, however, more states may be considered in the underlying Markov chain. Aggregated traffic from several sources may furthermore be given as a superposition of semi-Markovian models, thus resulting in a consequently larger state space. Using known techniques for an interval-based analysis studied in previous work [20], a verification of these large is not feasible in every case.

In the present work, we propose possible enhancements to these methods, in order to be able to yield more precise results for the given SMP models, as well as to make in feasible to analyse larger SMP models in the first place. These approaches include a modification of the polynomial factorization method, discussed in Section 2. As an alternative to finding the roots of the characteristic polynomial using verified algorithms, we consider this as an instance of the polynomial eigenvalue problem. This way, we can apply dedicated verified eigenvalue algorithms included, for instance, in INTLAB [18]. This is discussed in Section 3.

Additionally, we present an application of the successive over-relaxation (SOR) technique to the verified Wiener-Hopf factorization described in Section 4, which is determined using a SMP extension of the Grassmann-Jain algorithm [5]. This way, the speed of convergence may be improved.

As part of an ongoing development effort [10], all these methods and their respective improvements are included in our integrated problem-solving environment, *InterVerdiKom* (Section 5), which is used to provide some numerical examples (Section 6).

2 Polynomial Factorization

We assume a given discrete-time SMP/G/1 queueing system (see [13]). The arrival process describing the inter-arrival times of events A_t is given as a semi-Markovian process with M states, which we denote as SMP(M). The forwarding capacity of the router is given as the time needed to process a single event, S_t. The variables S_t are independent and identically distributed. We assume the random variable distributions to be constant over time and having finite support, hence, the time index t may be omitted.

Since the analysis relies on the difference process U, it is not relevant whether we consider a classical queueing system as described or a time-slotted system. In the latter approach, the arrival process denotes the amout of incoming events at a particular time slot, the service process describes the amount of events processed in the time slot. For a classical system, we determine the difference process as $U = S - A$, in the second case, it is given as $U = A - S$.

The support of U is limited, thus $-g < U < h$. Let σ_t denote the state of the difference process at time t. The state-dependent distributions of the difference process $P(U = k, \sigma_t = i)$ and state transition probabilities are then given by the distributions

$$u_{ij}(k) = P(\sigma_{t+1} = j, U = k | \sigma_t = i)$$

and denoted as a polynomial matrix $\mathcal{U}(z)$ of the respective generating functions, $\mathcal{U}(z) = (u_{ij}(z))$ with $u_{ij}(z) = \sum_{k=-g}^{h} u_{ij}(k) \cdot z^k$.

The workload W_t of a given queueing system at a point in time t – in accordance with Lindley's equation [16] – is determined by a convolution of the workload at the previous slot and the difference process, that is, $W_t = \max(W_{t-1} + U, 0)$. The goal of the polynomial factorization method is to get a solution for this recursive relation. If $E(U) < 0$ holds true, the workload converges to a steady-state distribution. We define this distribution as:

$$w_i(k) := \lim_{t\to\infty} P(W_t = k, \sigma_t = i), \quad k \in \mathbb{N}_0, i \in \{1, \ldots M\}.$$

The value given by $w_i(k)$ may be interpreted as the probability of the arrival process being in state i and the queue having a discrete workload of k. Let the distributions be denoted as a vector $\vec{w}(k) = (w_1(k), \ldots, w_M(k))^T$. We describe the basic idea of the polynomial factorization approach to compute the actual distribution and point the curious reader to previous works for more details [12]. To do so, we are looking for a representation for the workload distribution that fulfills the recursive relation

$$w_i(k) = \sum_{j=1}^{M} \sum_{l=-g}^{h} w_j(k+l) u_{ji}(-l) \text{ for } i \in \{1, \ldots, M\} \text{ and } k \geq h \tag{1}$$

and the side constraints

$$w_i(k) = \sum_{j=1}^{M} \sum_{l=-g}^{k} w_j(k+l) u_{ji}(-l) \text{ for } i \in \{1, \ldots, M\} \text{ and } 1 \leq k \leq h-1$$

$$w_i(0) = \sum_{j=1}^{M} \sum_{l=0}^{g} w_j(l) \sum_{n=l}^{g} u_{ji}(-n) \text{ for } i \in \{1, \ldots, M\}.$$

Theorem 2.1 *The representation $\vec{w}(k) = \beta^k \gamma \vec{\alpha}$ is a solution for relation 1 if and only if $\beta \in \mathbb{C} \setminus \{0\}$ is a solution of the characteristic system equation*

$$\det(\mathcal{U}^T(z^{-1}) - I) = 0 \tag{2}$$

and for $\vec{\alpha} \in \mathbb{C}^m$ holds

$$(\mathcal{U}^T(\beta^{-1}) - I)\vec{\alpha} = \vec{0}. \tag{3}$$

To prove the above theorem, we insert into relation 1 and yield for all $k \geq h$:

$$\begin{aligned}
\vec{w}(k) &= \sum_{n=-h}^{g} \vec{u}^T(-n)\vec{w}(k+n) \\
\Leftrightarrow \beta^k \gamma \vec{\alpha} &= \sum_{n=-h}^{g} \vec{u}^T(-n)(\beta^{k+n}\gamma\vec{\alpha}) \\
\Leftrightarrow \vec{\alpha} &= \sum_{n=-h}^{g} \vec{u}^T(-n)(\beta^n \vec{\alpha}) \\
\Leftrightarrow \vec{0} &= \left(\sum_{n=-h}^{g} \beta^n \vec{u}^T(-n) - I \right) \vec{\alpha} \\
\Leftrightarrow \vec{0} &= \left(\mathcal{U}^T(\beta^{-1}) - I \right) \vec{\alpha}.
\end{aligned}$$

We can safely assume that $\gamma \neq 0$, because $w(k) = 0$ for all $k \in \mathbb{N}_0$ concludes otherwise. The last equation holds for all $\beta \in \mathbb{C} \setminus \{0\}$ with $\det(\mathcal{U}^T(z^{-1}) - I) = 0$.

To consider the side constraints, we extend the representation to:

$$\vec{w}(k) = \sum_{\mu=1}^{Mh} \beta_\mu^k \gamma_\mu \vec{\alpha}_\mu \text{ for } k \in \mathbb{N}_0.$$

Definition 2.1 *The* characteristic polynomial *of a SMP/G/1 system is defined as*

$$p(z) := z^{Mh} \det(\mathcal{U}^T(\beta^{-1}) - I).$$

It is easy to see that the Mh roots of the characteristic polynomial fulfill the requirements of the theorem above. Given the root β_μ, $\mu = 1, \dots Mh$, we are able to compute the coefficient vector $\vec{\alpha_\mu}$ and the values γ_μ, by solving equation systems. Given 3 and $\sum_{i=1}^{M} \alpha_\mu^{(i)} = 1$, we solve

$$\begin{pmatrix} U_{1,1}(\beta_\mu^{-1}) - 1 & \cdots & U_{M-1,1}(\beta_\mu^{-1}) & U_{M,1}(\beta_\mu^{-1}) \\ \vdots & \ddots & \vdots & \vdots \\ U_{1,M-1}(\beta_\mu^{-1}) & \cdots & U_{M-1,M-1}(\beta_\mu^{-1}) - 1 & U_{M,M-1}(\beta_\mu^{-1}) \\ 1 & \cdots & 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_\mu^{(1)} \\ \vdots \\ \alpha_\mu^{(M-1)} \\ \alpha_\mu^{(M)} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

with $U_{ij}(z)$ denoting the respective elements of the polynomial matrix $\mathcal{U}(z)$. The coefficients γ_μ are derived from

$$\begin{pmatrix} \beta_1^{-1}\alpha_1^{(1)} & \cdots & \beta_{Mh}^{-1}\alpha_{Mh}^{(1)} \\ \vdots & \ddots & \vdots \\ \beta_1^{-1}\alpha_1^{(M)} & \cdots & \beta_{Mh}^{-1}\alpha_{Mh}^{(M)} \\ \beta_1^{-2}\alpha_1^{(1)} & \cdots & \beta_{Mh}^{-2}\alpha_{Mh}^{(1)} \\ \vdots & \ddots & \vdots \\ \beta_1^{-h+1}\alpha_1^{(M)} & \cdots & \beta_{Mh}^{-h+1}\alpha_{Mh}^{(M)} \\ \frac{1}{1-\beta_1}\alpha_1^{(1)} & \cdots & \frac{1}{1-\beta_{Mh}}\alpha_{Mh}^{(1)} \\ \vdots & \ddots & \vdots \\ \frac{1}{1-\beta_1}\alpha_1^{(M)} & \cdots & \frac{1}{1-\beta_{Mh}}\alpha_{Mh}^{(M)} \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{Mh} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ p_1 \\ \\ p_M \end{pmatrix}$$

with $p_i, i \in \{1, \dots M\}$ denoting the stationary probabilities of the difference process being in state i. These can be computed from

$$\sum_{i=1}^{M} p_i = 1 \text{ and } p_i = \sum_{j=1}^{M} p_j P(\sigma_{t+1} = j | \sigma_t = i).$$

From an algorithmic point of view, we have to compute a determinant of a polynomial matrix, find the roots of the resulting polynomial, and solve two equation systems to obtain a representation of the workload distribution. Considering the numerical stability of the approach, the most critical step is the correct computation of the polynomial roots.

When it comes to large models in terms of state space and distribution steps, the degree of the characteristic polynomial gets high as well. Because only the roots within the unit circle are relevant, we face the problem of selecting the correct roots amongst the floating-point approximations. In order to do so, we apply interval arithmetic: Starting from the approximations, we enclose the parameters of the model in point intervals and use the root verification algorithm from the C-XSC toolbox [9]. This way, we yield tight verified enclosures of the polynomial roots and are therefore able to determine the correct ones within the complex unit circle.

However, we still encounter problems in larger examples: If the interval enclosures are not disjunct or if they intersect the unit circle, there is no way of telling which ones are the correct roots that fulfill the theorem above. Hence, the ability to get tighter enclosures of the roots of the characteristic polynomial is an important objective for improvements.

3 Verified Eigenvalue Method

Looking for an improvement to the above polynomial factorization approach, we have studied eigenvalue methods to determine the parameters of the workload representation. As mentioned, the values β_μ are starting points for the workload analysis. In the following, we present an alternate approach of determining these values. As an additional benefit, we are able to compute the parameter vectors $\vec{\alpha}_\mu$ as well. Compared to the traditional factorization method, the new approach allows to compute these values directly from the polynomial matrix $\mathcal{U}(z)$ instead of first computing the determinant and the roots and the subsequent verification step.

Theorem 2.1 provides the foundation for the polynomial factorization. We are looking for the values β and $\vec{\alpha}$ that fulfill the relation $(\mathcal{U}^T(\beta^{-1}) - I)\vec{\alpha} = \vec{0}$ and, if $\vec{\alpha} \neq \vec{0}$, consequently $\det(\mathcal{U}^T(\beta^{-1}) - I) = 0$. Let $\lambda := \beta^{-1}$ and $\Psi(\lambda) := (\mathcal{U}^T(\beta^{-1}) - I)$, then finding the solution of the equation is equivalent to the *polyomial eigenvalue problem* (cf. [2]), that is, finding the complex eigenvalues λ and eigenvectors α that fulfill

$$\Psi(\lambda)\vec{\alpha} = \vec{0}.$$

The matrix polynomial $\Psi(\lambda)$ can be denoted in the form

$$\Psi(\lambda) = \lambda^l C_l + \lambda^{l-1} C_{l-1} + \ldots + \lambda C_1 + C_0,$$

with C_i being the particular coefficient matrices. By introducing block matrices A and B with

$$A := \begin{pmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & I \\ -C_0 & -C_1 & -C_2 & \cdots & -C_{l-1} \end{pmatrix}, \quad B := \begin{pmatrix} I & & & & \\ & I & & & \\ & & \ddots & & \vdots \\ & & & I & \\ & & & & C_l \end{pmatrix}$$

the polynomial eigenvalue problem can be linearized to $Az = \lambda Bz$, so that we can apply established eigenvalue solvers to compute solutions for λ. The eigenvectors α are derived from

$$z = (\vec{\alpha}^T, \lambda\vec{\alpha}^T, \ldots \lambda^{l-1}\vec{\alpha}^T)^T.$$

When performing the actual computation using floating-point arithmetic, we use the column of the matrix z that results in the smallest numerical error, that is, we evaluate $\Psi(\lambda)\vec{\alpha}$ for each column and choose the values $\vec{\alpha}$ resulting in the smallest magnitude.

From the algorithmic point of view, we have to perform a linearization of the polynomial matrix, apply an eigenvalue solver and solve one equation system for the parameters γ. In our implementation, we apply the eigenvalue solving routines in the LAPACK package to determine approximate values λ. We have ported the *verifyeig()*

routine from S. Rump's INTLAB package [18] to C++ to get verified enclosures for the approximations.

However, we may face a slight problem when computing the eigenvalues: If the characteristic polynomial has roots at zero, these roots correspond to eigenvalues at zero, meaning that $\lambda = \beta^{-1}$ is a division by zero. To cope with this situation, we reverse the order of the coefficient matrices of the polynomial matrix $\mathcal{U}(z)$ to determine the values β directly.

4 Wiener-Hopf-Factorization

Another approach to perform a workload analysis of a given SMP/G/1 queueing system is by Wiener-Hopf factorization using a modified Grassmann-Jain algorithm. W. Grassmann and L. Jain first applied this method to perform a workload analysis on a GI/G/1 queueing system. They described three variants of their algorithm, and proved the convergence of the slowest variant [5].

It is possible to extend the algorithm to determine the Wiener-Hopf factorization of SMP/G/1 queues, thereby providing another means for workload analysis [7]. However, a formal proof of convergence for the algorithm is not given. We apply interval arithmetic to provide a computer-assisted proof of convergence on a by-case basis in an additional verification step of the Wiener-Hopf factorization algorithm, the complete verified method is included in our integrated toolkit. In the following, we explain the basic idea of the algorithm and the subsequent verification step.

The idea is to divide the development of the workload of a given queueing system over time in busy phases and idle phases. A busy period is defined as the timespan between the first arrival, given an empty system with workload 0 until the system has processed the current request and possible additional arrivals, and runs empty again. The time spans between the busy periods are defined as idle periods. While a particular request is being served, additional arrivals may occur. These arrivals start another busy phase with the niveau of the current workload (a busy phase is said to be of niveau k if the workload at its beginning is k). A busy phase within a busy period may be seen as a busy period shifted to a particular niveau. From this point of view, a busy period is nothing else then a busy phase at niveau 0.

The basic idea for the Wiener-Hopf factorization is to split the given difference distribution into two distributions depending on each other and the original difference distribution. As in the previous section, let $U = A - S$ denote the difference of arrival and service times, $\sigma \in \{1, \ldots, M\}$ the states of the Markov chain embedded in the arrival process and

$$u_{ij}(k) := P(U_{n+1} = k, \sigma_{n+1} = j | \sigma_n = i);$$
$$i, j \in \{1, \ldots, M\}; \mathbf{u}(k) = (u_{ij}(k)), -g \leq k \leq h,$$

then $l_{ij}(k) = P(I = k, \sigma_E = j | \sigma_A = i)$ denotes the probability for an idle period of duration k, beginning in state σ_A and ending in state σ_E. As a second distribution, we introduce $v_{ij}(k)$ which describes the probability of a busy phase with niveau k and initial state j occurring in a busy period with initial state i. Analogous to the difference distribution $u_{ij}(k)$, they are denoted in matrix form $\mathbf{l}(k)$ and $\mathbf{v}(k)$. We get the following relations:

$$\mathbf{v}(i) = \mathbf{u}(i) + \sum_{j=1}^{h-i} \mathbf{v}(i+j)\mathbf{l}(j) \text{ if } i = h, \ldots, 0, \quad \mathbf{v}(i) = 0 \text{ if } i > h$$

$$\mathbf{l}(i) = \mathbf{u}(-i) + \sum_{j=0}^{g-i} \mathbf{v}(j)\mathbf{l}(i+j) \text{ if } i = g, \dots, 1, \quad \mathbf{l}(i) = 0 \text{ if } i > g$$

To compute the actual distributions, we start with an approximation for $l_{ij}^{(0)}(k)$:

$$l_{ij}(k) = \frac{u_{ij}(k)}{\sum_{n=1}^{g} u_{ij}(n)}, \quad k = 1, \dots g.$$

Furthermore, we have to handle the implicity of the above relation system with:

$$\mathbf{l}(i) = (I - \mathbf{v}(0))^{-1} \left[\mathbf{u}(-i) + \sum_{j=1}^{\min(g-i,h)} \mathbf{v}(j)\mathbf{l}(i+j) \right], \quad i = g, \dots 1.$$

Starting with the initial approximation $l_{ij}^{(0)}(k)$, we apply the relations above to calculate

$$l_{ij}^{(0)}(k) \to \dots \to l_{ij}^{(n)}(k) \to v_{ij}^{(n)}(k) \to l_{ij}^{(n+1)}(k) \to v_{ij}^{(n+1)}(k) \to \dots$$

until $\max_{i,j,k}(|l_{ij}^{(n+1)}(k) - l_{ij}^{(n)}(k)|) < \epsilon$ for a given threshold ϵ. Please note that the convergence of this algorithm is not formally proven, instead, we perform an additional verification step using interval arithmetic.

To do so, we start with the approximations determined by the outcome of the algorithm above, denoted as $\tilde{\mathbf{v}}(k)$ and $\tilde{\mathbf{l}}(k)$. We choose a parameter $\delta > \epsilon$ and set the intervals

$$[\mathbf{l}_{ij}^{(0)}(k)] = [\tilde{\mathbf{l}}_{ij}(k) - \delta, \tilde{\mathbf{l}}_{ij}(k) + \delta] \text{ and } [\mathbf{v}_{ij}^{(0)}(k)] = [\tilde{\mathbf{v}}_{ij}(k) - \delta, \tilde{\mathbf{v}}_{ij}(k) + \delta].$$

Using these enclosures, we apply the interval variants of the usual relations and perform a single step of the iteration above. If

$$[\mathbf{l}_{ij}^{(n+1)}(k)] \subseteq [\mathbf{l}_{ij}^{(n)}(k)]$$

holds true for all i, j, k, Brouwer's fixed point theorem guarantees a fixed point within the newly computed intervals, that is, the correct Wiener-Hopf factorization distributions are guaranteed to be included in the respective intervals. The interval diameter gives a measure on the accuracy of the actual interval-valued approximation. To increase the accuracy, further iteration steps may be performed as needed.

Next, we show how these distributions can be used to compute a workload distribution for the given queueing system. We denote the distribution $v_{ij}(k)$ using a generating function as $V_{ij}(z) = \sum_{k=0}^{h} v_{ij}(k)z^k$ and a polynomial matrix $\mathcal{V}(z) = (V_{ij}(z))$. The expectation value of arrivals during a busy period with initial state i, $E(N_i)$, is recursively given as

$$E(N_i) = 1 + \sum_{j=1}^{M} \sum_{k=0}^{h} v_{ij}(k)E(N_j) \Leftrightarrow (E(N_1), \dots, E(N_M))^T = (I - \mathcal{V}(1))^{-1}$$

The workload distribution in state i, denoted as a generating function $W_i(z)$, adheres to a similar relation

$$W_i(z)E(N_i) = 1 + \sum_{j=1}^{M} V_{ij}(z)W_j(z)E(N_j)$$

$$\Leftrightarrow (I - \mathcal{V}(z)) \left(E(N_1)\mathcal{W}_1(z), \ldots, E(N_M)\mathcal{W}_M(z)\right)^T = \vec{1}$$

By evaluating the relation above at $z = 0$, we yield the values $w_1(0), \ldots, w_M(0)$. To determine the higher workload probabilities, we derive the relation, leading to

$$E(N_i)w_i(n) = \sum_{j=1}^{M} \sum_{k=0}^{n} v_{ij}(k)E(N_j)w_j(n-k).$$

Starting with the probabilities for workload 0, we are able to compute all workload probabilities, depending on the state of the system. The overall workload probabilities - independent of the state of the semi-Markov process - are given by taking the probability of the initial state of a busy period l_i and the expectation value of arrivals into account:

$$w(n) = \frac{\sum_{i=1}^{M} l_i E(N_i) w_i(n)}{\sum_{i=1}^{M} l_i E(N_i)}$$

l_i is derived from $l_{ij} := \sum_{k=1}^{g} l_{ij}(k)$, with $l_j := \sum_{i=1}^{M} l_i l_{ij}, \sum_{j=1}^{M} l_j = 1$.

This method, compared to the polynomial factorization approach, shows some favorable properties. In particular, we are able to analyse models that are larger in terms of number of states and discretization steps of the state-specific dsitribution. However, a high level of autocorrelation in the SMP arrival process model, as it is found, for instance, in video traffic, may significantly slow down the convergence. Therefore, we investigated techniques to speed up the initial approximation step by a modification of the iteration very similar to successive overrelaxation (SOR). To do so, we introduce a coefficient ω into the iteration equations:

$$\mathbf{v}^{(n+1)}(i) = \mathbf{v}^{(n)}(i) + \omega \left[\mathbf{u}(i) + \sum_{j=1}^{h-i} \mathbf{v}^{(n)}(i+j)\mathbf{l}^{(n)}(j) - \mathbf{v}^{(n)}(i) \right]$$

$$\mathbf{l}^{(n+1)}(i) = \mathbf{l}^{(n)}(i) + \omega \left[\left(I - v^{(n)}(0)\right)^{-1} \left(\mathbf{u}(-i) + \sum_{j=0}^{g-i} \mathbf{v}^{(n)}(j)\mathbf{l}^{(n)}(i+j) \right) - \mathbf{l}^{(n)}(i) \right]$$

Choosing $\omega := 1$ results in the known relations. However, choosing $\omega > 1$ may speed up the iteration in some cases. We apply an adaptive algorithm to determine values for ω (see also [4]):

1. All j iteration steps, compute
$$q := \max_{i,j,k} \frac{|X_{ij}^{(n+1)}(k) - X_{ij}^{(n)}(k)|}{|X_{ij}^{(n)}(k) - X_{ij}^{(n-1)}(k)|}$$
2. If $q < 1$, set
$$q := \max(q, \omega - 1)$$
and compute a new ω:
$$\omega := \frac{1}{1 + \sqrt{1 - \frac{1}{q}\left(\frac{q+\omega-1}{\omega}\right)^2}}$$

However, the convergence of the iteration is not proven. During our experiments, we saw that a large value for ω may result in a divergent iteration.

5 Integrated Problem-Solving Environment

Our research has shown a clear interdependence between modeling and verified analysis of data traffic. On the one hand, the larger the semi-Markov models to be analyzed are in terms of state space and distribution steps, the more difficult it gets to provide a verification of the Wiener-Hopf or polynomial factorization. On the other hand, larger models provide a more accurate representation of the autocorrelation of an original data source, which is very important for an accurate analysis of the resulting queueing system [11]. Furthermore, the analysis methods presented in this work have different characteristics. While one method may be successful in providing a verified analysis of the workload, the other approaches may fail (see Section 6).

Therefore, our intention is to provide the interested user with a comprehensive toolkit that takes both modeling and analysis aspects into account. The methods described have been implemented as parts of our ongoing effort to create an integrated environment for modeling and verified analysis of correlated data traffic in communication networks, *InterVerdiKom*. In the following, we will explain some of the design principles of the toolkit. Some numerical examples in Section 6 will illustrate the advantages of the integrated approach for traffic analysis.

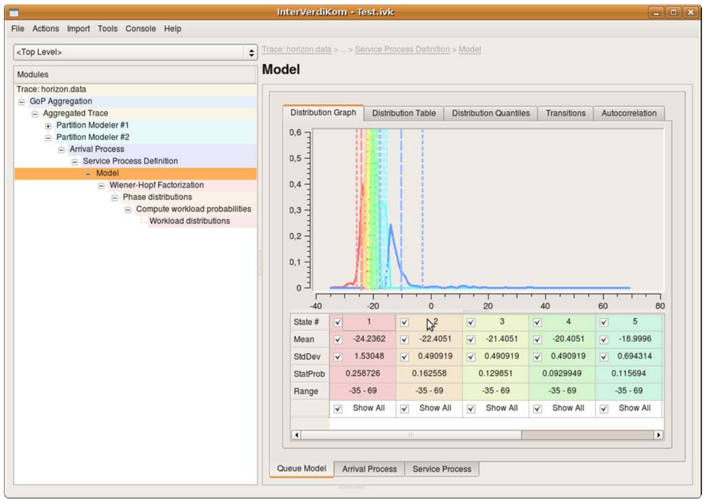
The analysis techniques included in the toolkit have been discussed earlier, the modeling methods are subject of a separate publication [11]. Our integrated approach includes two important aspects: On the one hand, this refers to a single data model used throughout the modeling and analysis process. This way, no conversions between different interval representations potentially introducing additional round-off errors are needed. Furthermore, the results from different analysis or modeling techniques are directly comparable.

On the other hand, we combine the modeling and subsequent analsis into a single workflow. This enables the user to modify the modeling/analysis process at different stages from within a consistent user interface to get the best results possible. Since there are several techniques applicable at certain points in the workflow, we chose to represent the process as a tree: At each node, the user can branch off, apply different techniques and compare the results. For instance, if a proof of convergence is not possible using different parameters of the Wiener-Hopf factorization, he may try polynomial factorization or the eigenvalue approach as well as change parameters of the modeling process to produce smaller semi-Markov models. Since the workflow is organized as a tree, the complete process and all results from the different parameter alternatives and algorithms remain traceable within the same environment.

To further support this approach, we provide the user with a consistent and convienient graphical user interface (Figure 1 displays a screenshot of the integrated environment). The tree representation of the workflow is shown at the left, details about the current step are given in the main view to the right. A console providing detailed information about the algorithms and possible difficulties is also included in the application, but not shown in the screenshot.

InterVerdiKom takes a number of other important aspects of modern software development into account: The toolkit shows a modular composition, making it easy to add additional views or numerical methods. The different algorithms may be run at the same time on multiprocessor machines. We have included a number of export options, for instance, to MATLAB and CSV files.

The application makes use of a number of external libraries, namely the C-XSC toolbox [9] and LAPACK [1] for the numerical computations and Qt for the graphical interface.



State #	✓ 1	✓ 2	✓ 3	✓ 4	✓ 5
Mean	✓ -24.2362	✓ -22.4051	✓ -21.4051	✓ -20.4051	✓ -18.9996
StdDev	✓ 1.53048	✓ 0.490919	✓ 0.490919	✓ 0.490919	✓ 0.694314
StatProb	0.258726	0.162558	0.129851	0.0929949	0.115694
Range	-35 - 69	-35 - 69	-35 - 69	-35 - 69	-35 - 69
	✓ Show All	✓ Show All	✓ Show All	✓ Show All	✓ Show All

Figure 1: Screenshot of the *InterVerdiKom* application

6 Examples

We illustrate the capabilities and limits of the presented techniques and their implementation in *InterVerdiKom* by two examples. The first one is given by a semi-Markov arrival process with three states. The transition probabilities are given by

$$P = \begin{pmatrix} 0.2 & 0.6 & 0.2 \\ 0.6 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.3 \end{pmatrix},$$

the state-specific distributions by

$$\begin{aligned} P(A_1 = 0) &= 0.01, & P(A_1 = k) &= 0.02 & &\text{for } k = 1, \ldots, 49, \\ P(A_1 = 50) &= 0.01, & & & & \\ P(A_2 = 0) &= 0.02, & P(A_2 = k) &= 0.04 & &\text{for } k = 1, \ldots, 24, \\ P(A_2 = 25) &= 0.02, & P(A_2 = k) &= 0 & &\text{for } k = 26, \ldots, 50, \\ P(A_3 = 0) &= 0.02, & P(A_3 = k) &= 0 & &\text{for } k = 1, \ldots, 25, \\ P(A_3 = 50) &= 0.02, & P(A_3 = k) &= 0.04 & &\text{for } k = 26, \ldots, 49. \end{aligned}$$

A selection of the resulting interval enclosures for different workload probabilities and the particular methods presented in this work are given in Table 1. Please note that our implementation of the verified eigenvalue method currently does not employ circular complex interval arithmetic, hence, the accuracy of the interval enclosures is significantly less than when applied with Rump's original implementation in INTLAB due to the wrapping effect. As part of our current work, we intend to make use of circular arithmetic as well.

w	Polynomial Factorization	Verified Eigenvalues
$w(0)$	$9.34200080^{80603523}_{73357808} \cdot 10^{-1}$	$9.342000^{8196924254}_{7957037043} \cdot 10^{-1}$
$w(10)$	$7.883270671^{9369095}_{0112136} \cdot 10^{-5}$	$7.8832706^{868932508}_{560549170} \cdot 10^{-5}$
$w(50)$	$1.146819437^{1166538}_{0399568} \cdot 10^{-17}$	$1.14681943^{84919901}_{56646422} \cdot 10^{-17}$

w	Wiener-Hopf Factorization
$w(0)$	$9.34200080769^{84362}_{76924} \cdot 10^{-1}$
$w(10)$	$7.88327067147^{43644}_{37587} \cdot 10^{-5}$
$w(50)$	$1.146819437078^{3589}_{2563} \cdot 10^{-17}$

Table 1: Interval enclosures for workload probabilities, SMP(3) example

w	Wiener-Hopf Factorization
$w(0)$	$9.87362101073^{77165}_{68472} \cdot 10^{-1}$
$w(10)$	$5.64049554820^{34587}_{23029} \cdot 10^{-4}$
$w(50)$	$3.74304807822^{96515}_{88697} \cdot 10^{-6}$

Table 2: Interval enclosures for workload probabilities, video traffic model

For the second example, we have generated a SMP arrival model with 7 states for a given video trace with high autocorrelation, the state-specific distributions are modeled in 70 steps. The service process is given as a discrete renewal model with a constant capacity $E(S) = E(A) + 4\sigma(A)$. We apply the same discretization technique like for the arrival process (see [11]).

In this more complex example, an analysis using the polynomial factorization method and the verified eigenvalue technique is not feasible due to numerical errors. In particular, a selection of the correct roots or eigenvalues, respectively, is not possible because of inaccuracies. Hence, we give the results for the Wiener-Hopf factorization only.

7 Conclusion and further work

In this work, we have presented three different methods for the verified interval-based analysis of semi-Markovian queueing systems. All of these techniques are included in our toolkit InterVerdiKom which provides an integrated approach to modeling and verified analysis of correlated traffic in service-integrated communication networks.

We have seen that the particular algorithms have different capabilities. The choice of which algorithm to apply depends on the problem at hand, however, the Wiener-Hopf approach, in general, provides the most accurate results and is therefore appropriate for the analysis of larger SMP models resulting, for instance, from modeling of video traffic.

Our integrated environment allows the user to branch off at each stage of the modeling and analysis workflow and apply alternative techniques, while remaining in the same environment. A conversion of the interval-valued intermediate results is avoided, thus increasing the accuracy of the overall result and making the results directly comparable.

We intend to change our implementation of the *verifyeig* algorithm to make use of circular interval arithmetic to reduce inaccuraries introduced by the wrapping effect. This way, we expect the accuracy of the proposed verified eigenvalue technique to increase. Furthermore, we want to enhance *InterVerdiKom* by adding techniques for describing correlated output processes and superposition of arrival processes of the queueing systems under consideration. This will enable an integrated interval-valued analysis of open queueing networks.

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK User's Guide*. SIAM, third edition edition, 1999.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, 2000.

[3] B. Carpenter and K. Nichols. Differentiated Services in the Internet. In *Proceedings of the IEEE*, volume 90, pages 1479–1494, 2002.

[4] G. Engeln-Müllges, K. Niederdrenk, and R. Wodicka. *Numerik-Algorithmen*. Springer, 2005.

[5] W. K. Grassmann and J. L. Jain. Numerical Solutions of the Waiting Time Distribution and Idle Time Distribution of the Arithmetic GI/G/1 Queue. *Operations Research*, 37:141–150, 1989.

[6] G. Hasslinger. Semi-Markovian Modelling and Performance Analysis of Variable Rate Traffic in ATM Networks. *Telecommunication Systems*, 7:281–298, 1997.

[7] G. Hasslinger. Waiting Time, Busy Periods and Output Models of a Server Analyzed via Wiener-Hopf Factorization. *Performance Evaluation*, 40(1-3):3–26, 2000.

[8] G. Hasslinger and P. Takes. Real Time Video Traffic Characteristics and Dimensioning Regarding QoS Demands. In *Proceedings of the 18th International Teletraffic Congress*, 2003.

[9] W. Hofschuster and W. Krämer. C-XSC 2.0: A C++ Library for Extended Scientific Computing. In René Alt, Andreas Frommer, R. Baker Kearfott, and Wolfram Luther, editors, *Numerical Software with Result Verification*, volume 2991 of *Lecture Notes in Computer Science*, pages 15–35. Springer, 2004.

[10] S. Kempken. A Workload Analysis Tool for Discrete-Time Semi-Markovian Servers. In *Proceedings of the 12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN)*, 2006.

[11] S. Kempken, G. Hasslinger, and W. Luther. Parameter Estimation and Optimization Techniques for Discrete-Time Semi-Markov Models of H.264/AVC Video Traffic. *Telecommunication Systems*, 2008.

[12] S. Kempken, W. Luther, and G. Hasslinger. A Tool for Verified Analysis of Transient and Steady States of Queues. In *Proceedings from the 2006 workshop on Tools for solving sturctured Markov chains (VALUETOOLS)*, volume 201 of *ACM International Conference Proceedings*. ACM, 2006.

[13] L. Kleinrock. *Queueing Systems.* Wiley, 1975.

[14] M. Krunz and A. Ramasamy. The Correlation Structure for a Class of Scene-Based Video Models and Its Impact on the Dimensioning of Video Buffers. *IEEE Transactions on Multimedia*, 2(1):27–36, 2000.

[15] San-Qi Li. A General Solution Technique for Discrete Queueing Analysis of Multi-Media Traffic on ATM. *IEEE Transactions on Communications*, 39(7):1115–1132, 1991.

[16] D. V. Lindley. The Theory of Queues with a Single Server. *Proceedings of the Cambridge Philosophic Society*, 48:277–289, 1952.

[17] O. Rose. Simple and Efficient Models for Variable Bit Rate MPEG Video Traffic. *Performance Evaluation*, 30:69–85, 1997.

[18] S. M. Rump. Computational error bounds for multiple or nearly multiple eigenvalues. *Linear Algebra and its Applications*, 324:209–226, 2001.

[19] P. Salvador, R. Valadas, and A. Pacheco, Multiscale Fitting Procedure Using Markov Modulated Poisson Processes. *Telecommunication Systems*, 23(1-2):123–148, 2003.

[20] S. Kempken, W. Luther, and G. Haßlinger. Reliable computation of workload distributions using semi-Markov processes, In: *Proceedings of the 13th International Conference on Analytical and Stochastic Modelling Techniques and Applications ASMTA*, pages 111–116, 2006.

[21] J. Wroclawski, *The Use of RSVP with IETF Integrated Services*, Proposed IETF Standard, RFC 2210, 1997.