Matrix-geometric Solutions of M/G/1 Type Markov Chains: A Unifying Generalized State-space Approach

Nail Akar
Technology Planning & Integration
Sprint
9300 Metcalf Avenue
Overland Park, KS 66212
akar@sprintcorp.com

Nihat Cem Oğuz and Khosrow Sohraby*
Computer Science Telecommunications
University of Missouri-Kansas City
5100 Rockhill Road
Kansas City, MO 64110
{ncoguz,sohraby}@cstp.umkc.edu

Abstract

In this paper, we present an algorithmic approach to find the stationary probability distribution of M/G/1 type Markov chains which arise frequently in performance analysis of computer and communication networks. The approach unifies finite- and infinite-level Markov chains of this type through a generalized state-space representation for the probability generating function of the stationary solution. When the underlying probability generating matrices are rational, the solution vector for level k, x_k, is shown to be in the matrix-geometric form $x_{k+1} = gF^kH$, $k \geq 0$, for the infinite-level case whereas it takes the modified form $x_{k+1} = g_1F_1^kH_1 + g_2F_2^{K-k-1}H_2$, $0 \leq k < K$, for the finite-level case. The matrix parameters in the above two expressions can be obtained by decomposing the generalized system into *forward* and *backward subsystems*, or equivalently, by finding bases for certain *generalized invariant subspaces* of a regular pencil $\lambda E - A$. We note that the computation of such bases can efficiently be carried out using advanced numerical linear algebra techniques including *matrix-sign function* iterations with quadratic convergence rates or *ordered generalized Schur decomposition*. The simplicity of the matrix-geometric form of the solution allows one to obtain various performance measures of interest easily, e.g., overflow probabilities and the moments of the level distribution, which is a significant advantage over conventional recursive methods.

Index Terms: Performance analysis of high-speed communication networks, ATM multiplexer analysis, M/G/1 type Markov chains, polynomial matrix fractional descriptions, generalized difference equations, generalized invariant subspaces, matrix-sign function, generalized Schur decomposition.

*This work was partially supported by DARPA/ITO under grant A0-F316 and by NSF under grant NCR-950814.

1 Introduction

In this paper, we study Markov chains of M/G/1 type with finite or infinite number of levels. The state-space of an infinite-level (or simply infinite) M/G/1 type Markov chain consists of integer pairs (i, j) where i, the *level* of the chain, takes on an infinite set of values ($i \geq 0$) and j, the *phase* of the chain, takes on a finite set of values ($0 \leq j < m$). The transition probability matrix of this chain has the block-partitioned form [29]

$$P = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & \cdots \\ A_0 & A_1 & A_2 & A_3 & \cdots \\ & A_0 & A_1 & A_2 & \cdots \\ & & A_0 & A_1 & \cdots \\ & & \vdots & \vdots & \ddots \end{bmatrix}, \tag{1}$$

where A_i and B_i, $i \geq 0$, are $m \times m$ matrices. Assuming that P is irreducible and positive recurrent, we find the stationary probability vector $x = [\begin{array}{ccc} x_0 & x_1 & \cdots \end{array}]$ which satisfies

$$x = xP, \quad xe = 1, \tag{2}$$

where x_i, $i \geq 0$, is $1 \times m$, and e is an infinite column vector of ones.

When the number of levels is finite, say $K + 1$, the transition probability matrix takes the block upper-Hessenberg form

$$P = \begin{bmatrix} B_0 & B_1 & B_2 & \cdots & B_{K-1} & \bar{B}_{K-1} \\ A_0 & A_1 & A_2 & \cdots & A_{K-1} & \bar{A}_{K-1} \\ & A_0 & A_1 & \cdots & A_{K-2} & \bar{A}_{K-2} \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & A_0 & A_1 & \bar{A}_1 \\ & & & & A_0 & \bar{A}_0 \end{bmatrix}, \tag{3}$$

where $\bar{A}_i$, $0 \leq i < K$, and $\bar{B}_{K-1}$ are $m \times m$, and constitute the boundary at level K. We then study the solution vector $x = [\begin{array}{cccc} x_0 & x_1 & \cdots & x_K \end{array}]$ which satisfies (2) with e this time being a column vector of ones of length $m(K + 1)$. Throughout the paper, e will denote a column vector of ones of suitable size.

Both infinite and finite M/G/1 type Markov chains arise frequently in the performance analysis of ATM (Asynchronous Transfer Mode) networks. In an ATM network, the basic unit of information is a fixed-length *cell* and the sharing of common network resources (bandwidth, buffers, etc.) among virtual connections is made on a statistical multiplexing basis. Statistical quality of service guarantees are integral to an ATM network, necessitating accurate traffic and performance analysis tools to determine the cell loss rate, cell delay, and cell delay variation in an ATM node (switch, multiplexer, etc.). This is in general difficult due to multiplexing of typically a large number of connections and burstiness of individual cell streams at possibly different time scales. One popular approach is to approximate such complex non-renewal input processes by analytically tractable Markovian models either at the connection [34], [10] or at the link (physical or logical) level [18], [19], [27]. Markovian Arrival Process (MAP) [26] and Batch Markovian Arrival Process (BMAP) [24] have been used extensively in ATM performance evaluation in continuous-time. For example, the well-known Markov-Modulated Poisson Process (MMPP) is a sub-case of MAP [18]. Various other Markovian traffic models including Markov-Modulated Bernoulli Process (MMBP) or its generalization Discrete Batch Markovian Arrival Process (DBMAP) are also used to model the correlated nature of ATM traffic streams in discrete-time [34], [35]. We note that DBMAP model allows *batch* arrivals in one cell-time [30] and is suitable for modeling aggregate traffic. Such processes in continuous- or discrete-time when fed into a single-server queue are known to give rise to M/G/1 type Markov chains [29], where the phase of the chain represents the state of the underlying Markovian model that governs (or modulates) the arrivals, and the level of the chain represents the queue length.

While the infinite M/G/1 chain seems to lack physical justification due to limited storage capacities in ATM nodes, it usually serves as an efficient approximation to the case of finite but large number of levels. Infinite M/G/1 models have especially been used in the analysis of asymptotic queue length behavior which is closely linked with effective bandwidth computations for call admission control in ATM networks [34], [35], [10]. Assuming an output buffer capacity of K cells, the infinite M/G/1 chain can be truncated at level K to obtain a finite M/G/1 chain of the form (3). Assuming no particular buffer management scheme in effect, this truncation is generally done by writing the boundary at level K as

$$\bar{A}_i \triangleq \sum_{j=i+1}^{\infty} A_j, \;\; 0 \leq i < K, \quad \text{and } \bar{B}_{K-1} \triangleq \sum_{j=K}^{\infty} B_j. \tag{4}$$

On the other hand, the boundary behavior at level 0 is generally captured by defining

$$B_0 \triangleq A_0 + A_1 \text{ and } B_i \triangleq A_{i+1}, \;\; 1 \leq i < K, \tag{5}$$

if the node can forward an incoming cell without any delay. In the case that an incoming cell is subject to one cell-time delay even when the buffer is empty, one has

$$B_i \triangleq A_i, \;\; 0 \leq i < K. \tag{6}$$

Other possibilities for the boundary sequence $\{B_i\}$ also exist [29].

For the solution of infinite and finite M/G/1 chains, we take an algebraic approach which is entirely different than the conventional methods. This technique unifies finite and infinite models, and consists of obtaining a generalized state-space representation of the probability generating function of the stationary solution. The generalized system is then decomposed into its forward and backward subsystems which in turn result in a matrix-geometric solution for infinite M/G/1 chains:

$$x_{k+1} = gF^kH, \;\; k \geq 0. \tag{7}$$

Using the same generalized system and its forward-backward decomposition, we further show that the solution vector for level k for finite M/G/1 chains is expressible as

$$x_{k+1} = g_1F_1^kH_1 + g_2F_2^{K-k-1}H_2, \;\; 0 \leq k \leq K-1. \tag{8}$$

The computational algorithm we propose to find the elements of the above matrix-geometric expressions is based on the *matrix-sign function iterations* [7] or the *generalized Schur decomposition with ordering* [20], leading to a method which is in general relatively faster than the conventional recursive algorithms, with less storage requirements. Besides, the simple compact form for the stationary probabilities substantially facilitates calculating certain performance measures of interest such as buffer overflow probabilities (or cell loss rates) and moments of the level distribution (or cell delay and cell delay variation). It also proves useful in the analysis of asymptotic queue length behavior.

The transition probability matrices of (1) and (3) are said to be in *canonical* form. Non-canonical chains with complex boundaries can also be studied in the same unifying generalized state-space framework. A case which was studied in [3] is the M/G/1 chain below with multiple boundary levels:

$$P = \begin{bmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} & \cdots \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \\ B_{N-1,0} & B_{N-1,1} & B_{N-1,2} & B_{N-1,3} & \cdots \\ A_0 & A_1 & A_2 & A_3 & \cdots \\ 0 & A_0 & A_1 & A_2 & \cdots \\ 0 & 0 & A_0 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{9}$$

where $B_{i,j}$ and A_j, $j \geq 0$, $0 \leq i \leq N-1$, are $m \times m$ matrices and N denotes the number of boundary levels. When $N = 1$, the probability model reduces to the canonical form (1). We show in [3] that the solution vector for level $k \geq N$ has the simple matrix-geometric form

$$x_{k+N} = gF^k H, \;\; k \geq 0. \tag{10}$$

Using invariant subspace computations in the solution of infinite M/G/1 and G/M/1 type Markov chains has been proposed before in [2]. In [3], this approach has been refined in the generalized state-space framework to eliminate recursive computations traditionally required to find the stationary probabilities of an infinite M/G/1 chain. The current paper is an extended version of [3], and presents the unifying generalized state-space approach for the stationary solution of infinite/finite and single-/multiple-boundary M/G/1 type chains arising in the performance analysis of computer and communication systems. Furthermore, we introduce the *ordered generalized Schur decomposition* in this paper as the numerical engine that implements the generalized state-space approach as well as the *matrix-sign function* method which was studied extensively in [2] and [3]. Based on the numerical experiments we have performed, the former method appears to outperform the serial version of the latter in terms of execution times and accuracy. However, we note that the matrix-sign function iterations are parallelizable at the algorithm level, and significant execution time reductions can potentially be attained by means of parallel implementations [15].

The paper is organized as follows. In Section 2, we introduce the generalized state-space approach for solving infinite M/G/1 type Markov chains. Section 3 describes the two algorithms that implement this approach; one algorithm is based on the matrix-sign function, and the other on the ordered generalized Schur decomposition. Then Sections 4 and 5 extend the formulation to also cover finite M/G/1 chains and the non-canonical case of multiple boundary levels, respectively. Numerical examples are provided in Section 6 to demonstrate the accuracy and efficiency of the approach.

2 Infinite M/G/1 Type Markov Chains

For the mathematical formulation of the problem, we first need to define the two d-domain probability generating matrices

$$A(d) \triangleq \sum_{i=0}^{\infty} A_i d^{-i} \quad \text{and} \quad B(d) \triangleq \sum_{i=0}^{\infty} B_i d^{-i}, \tag{11}$$

which are related to their z-domain counterparts as $A(d) = A(z)|_{z=d^{-1}}$ and $B(d) = B(z)|_{z=d^{-1}}$, respectively. We then make the assumption that the transform matrices $A(d)$ and $B(d)$ are rational, i.e., the entries of $A(d)$ and $B(d)$ are rational functions of d. This assumption is not restrictive because:

(i) Most of the probability models of M/G/1 type encountered in computer and communication systems naturally give rise to rational transform matrices.

(ii) When the transform matrices are general, conventional methods make use of truncation to replace the infinite matrix sequences $\{A_i\}$ and $\{B_i\}$ appropriately by finite sequences for computational tractability, and this amounts to approximating the transform matrices by rational matrices. Our model avoids truncation by taking advantage of the rational structure of $A(d)$ and $B(d)$, and thus generalizes the existing models.

(iii) It is in general advantageous to use rational functions to approximate general (possibly irrational) probability generating matrices. See for example [1] in which the deterministic service time in a MAP/D/1/K queue is approximated by Padé approximations in transform domain to successfully estimate the cell loss rates in an ATM multiplexer.

Under the above assumption, one can express $A(d)$ and $B(d)$ as a *stable right coprime polynomial matrix fraction:*

$$\begin{bmatrix} A(d) \\ B(d) \end{bmatrix} = \begin{bmatrix} P(d) \\ R(d) \end{bmatrix} Q^{-1}(d), \tag{12}$$

where $P(d)$, $R(d)$, and $Q(d)$ are $m \times m$ polynomial matrices of d [21], with polynomial degrees p, r, and q, respectively. We note that $A(d)$ and $B(d)$ are proper rational matrices, and hence, the relations $p \leq q$ and $r \leq q$ hold. Moreover, stable right coprimeness is imposed on the fraction to avoid redundancies in the matrix-fractional description, and implies that all the roots of $\det[Q(d)]$ lie in the open unit disk[1].

In the following, we first discuss how the fractional description of (12) can be obtained generically, and provide some teletraffic examples naturally yielding such descriptions. Then, after outlining a slightly modified version of the traditional iterative solution methods, we introduce the generalized state-space approach of this paper.

2.1 Obtaining Stable Right Coprime Fractions

Consider a stable proper transform matrix, $A(d)$, of size $p \times q$. One can generically obtain a stable right coprime fraction of $A(d)$ as follows. Let $l_i(d)$, $1 \leq i \leq q$, be the least common multiple of all the denominators of the i-th column entries of $A(d)$. Define $Q(d) \triangleq \text{diag}\{l_i(d)\}$ and $P(d) \triangleq A(d)Q(d)$. It is then clear that the fraction $A(d) = P(d)Q^{-1}(d)$ is a stable right coprime polynomial fraction. As an example, consider a two-state Markov-modulated geometric source. Let t_{ij} be the state transition probabilities of the modulating chain, and r_{ij} be the geometric rate parameter associated with the transitions. Then, the entries $a_{ij}(d)$ of $A(d)$ are given as

$$a_{ij}(d) = t_{ij}\frac{(1-r_{ij})d}{d-r_{ij}}, \quad i,j = 1,2.$$

If we assume that r_{ij}'s are all different, then the entries $q_{ij}(d)$ and $p_{ij}(d)$ of $Q(d)$ and $P(d)$ are found respectively as

$$q_{ii}(d) = (d-r_{1i})(d-r_{2i}), \quad i = 1,2,$$

and

$$p_{ij}(d) = t_{ij}(1-r_{ij})d(d-r_{kj}), \quad i,j = 1,2, \text{ and } k = 3-i.$$

For a wide variety of teletraffic models, however, one may not need to take this generic approach as the fractions can directly be obtained from the problem description. Below, we give three popular models from the teletraffic literature, and find a stable right coprime pair of matrices $P(d)$ and $Q(d)$ for the probability generating matrix $A(d)$. We also note that the fraction for $B(d)$ can generally be obtained through that for $A(d)$ easily as in (5) or (6).

1. *Quasi-birth-and-death processes [36], [28], [23]:* If in the structure of P in (1), state transitions are restricted to take place between adjacent levels only, the resulting model is called a quasi-birth-and-death process (QBD). That is, for QBD chains, $A_k = 0$ for $k > 2$, and

$$A(d) = A_0 + A_1 d^{-1} + A_2 d^{-2}.$$

The choice of

$$P(d) = A_2 + A_1 d + A_0 d^2 \quad \text{and} \quad Q(d) = d^2 I$$

[1] Our recent experiments indicate that the generalized state-space method works even when there are redundancies, that is, even when coprimeness is not sought. See Appendix A for a brief mathematical overview of stability and right coprimeness concerning polynomial matrices and polynomial matrix fractions.

gives a stable right coprime fraction for $A(d)$. Note that this formulation appropriately extend to obtain fractions for the more general case in which $A_k = 0$ for $k > N$ and $2 < N < \infty$.

2. *Single-server discrete-time queue with modulated arrivals:* Consider a discrete-time queue with a single server and with arrivals modulated by a finite-state discrete-time Markov chain [34]. Assume that the modulating chain has m states with transitions occurring at slot boundaries. Let t_{ij}, $0 \leq i, j \leq m-1$, denote the transition probabilities. Also let h_{ik} denote the probability of k arrivals when the modulating chain resides in state i. Assume that

$$h_i(d) = \sum_{k=0}^{\infty} h_{ik} d^{-k} \tag{13}$$

is a rational function of d (for example, a discrete phase-type distribution). Let the queue length and state of the modulating chain be associated with our level and phase definitions. If we write $h_i(d) = p_i(d)/q_i(d)$, $0 \leq i \leq m-1$, then $A(d)$ can be written as

$$A(d) = \underbrace{Q_0 \tilde{P}(d)}_{P(d)} Q^{-1}(d)$$

which is a stable right coprime fraction with $Q_0 = [t_{ij}]$, $\tilde{P}(d) = \mathrm{diag}\{p_i(d)\}$, and $Q(d) = \mathrm{diag}\{q_i(d)\}$.

3. *MMPP/G/1 queue [24]:* Consider a single-server queue with the arrival process modeled as a Markov-Modulated Poisson Process (MMPP) characterized by the infinitesimal generator matrix Λ of the underlying Markov chain and the rate matrix $R = \mathrm{diag}\{\lambda_i\}$, $0 \leq i \leq m-1$. We assume that the service time distribution H is Coxian, i.e., H has a rational Laplace-Stieltjes transform $h(\cdot)$, so that we can write $h(s) = p(s)/q(s)$ for some coprime polynomials p and q. Considering the embedded Markov renewal process at departure epochs, we obtain a Markov chain of M/G/1 type with

$$A(d) = h(\Lambda - R - \Lambda d^{-1}),$$

which is a rational function of d [24]. The polynomial fractions of $A(d)$ can directly be obtained as

$$Q(d) = d^w \, q(\Lambda - R - \Lambda d^{-1}) \quad \text{and} \quad P(d) = d^w \, p(\Lambda - R - \Lambda d^{-1}),$$

where w is the degree of polynomial $q(s)$.

2.2 Matrix-analytic Method

We now outline an efficient iterative method for finding the stationary solution as in (2) of an infinite M/G/1 type Markov chain. This method is based on the matrix-analytic approach pioneered by Neuts [29] with a slight modification to take advantage of the rationality of $A(d)$ (also see [25] for a similar approach for the BMAP/G/1 queue). In this method, the key is to find the unique minimal nonnegative solution G_* of the nonlinear matrix equation

$$G = \sum_{k=0}^{\infty} A_k G^k. \tag{14}$$

A successive substitution iteration for finding G_* exploiting the rationality of $A(d)$ to avoid truncation of this infinite summation is

$$G_0 = 0, \quad G_{j+1} = \tilde{Q}^{-1}(G_j)\tilde{P}(G_j), \quad j \geq 1, \tag{15}$$

where $\tilde{Q}^{-1}(d)\tilde{P}(d)$ is a polynomial fraction for $A(d^{-1})$, or equivalently, $A(z) = \tilde{Q}^{-1}(z)\tilde{P}(z)$ in z-domain. (note that this is a left polynomial fraction as opposed to (12); also see [2]). Previous numerical experiments indicate that this iteration has a linear convergence rate [2]. It is shown in [33] that x_0 is equal to the stationary probability vector κ of the stochastic matrix $\sum_{i=0}^{\infty} B_i G_*^i$ normalized as

$$x_0 = \frac{\kappa}{n}, \quad n = 1 + \frac{\kappa}{1-\rho}\left[\beta + [B(1) - I]\,[I - A(1) + e\pi]^{-1}\alpha\right] \tag{16}$$

where

$$\alpha = -A^{(1)}(1-)e, \ \beta = -B^{(1)}(1-)e, \quad \rho = \pi\alpha, \tag{17}$$

π is the stationary probability vector of $A(1)$, and the traffic parameter (or utilization) ρ is less than unity. Once x_0 is found, the vectors x_k, $k \geq 1$, can be obtained recursively by [31]

$$x_k = \left[x_0\hat{B}_k + \sum_{i=1}^{k-1} x_i\hat{A}_{k-i+1}\right](I - \hat{A}_1)^{-1}, \tag{18}$$

where

$$\hat{A}_k \triangleq \sum_{i \geq k} A_i G_*^{i-k}, \quad \hat{B}_k \triangleq \sum_{i \geq k} B_i G_*^{i-k}. \tag{19}$$

Note that computation of x_k, $k > 0$, as in (18) requires truncation of the infinite matrix sequences $\{A_i\}$ and $\{B_i\}$. Due to the low linear convergence rates of the successive substitution iterations to find G_* and depending on the truncation index required to attain a certain accuracy, the matrix-analytic approach may in general incur considerable execution times and storage requirements especially under heavy traffic conditions. The generalized state-space approach differs significantly from the matrix-analytic approach, and is presented in Section 2.4 after the following brief overview of invariant subspaces.

2.3 Overview of Invariant Subspaces

Here we give a brief description of ordinary and generalized invariant subspaces based mainly on [12] and [16]. We use the following notation. Uppercase is used for matrices and lowercase for vectors, both being defined over the field of real numbers $\mathbf{R}$. $\lambda(A)$ denotes the spectrum, i.e., the set of eigenvalues, of A. A constant, polynomial or rational matrix is called *regular* when it is square and has a nonzero determinant. Otherwise, it is called *singular*. A *subspace* $\mathcal{S}$ is a subset of $\mathbf{R}^m$ that is closed under the operations of addition and scalar multiplication. Im A denotes the image (or the column space) of A. $A\mathcal{S}$ is the image of $\mathcal{S}$ under A. An *invariant subspace* $\mathcal{S}$ of A satisfies $A\mathcal{S} \subset \mathcal{S}$ where $\subset$ denotes inclusion. $\mathcal{S} + \mathcal{T}$ and $\mathcal{S} \oplus \mathcal{T}$ are the sum and direct sum, respectively, of the subspaces $\mathcal{S}$ and $\mathcal{T}$. Let $\mathcal{S} \oplus \mathcal{T} = \mathbf{R}^m$ and assume that $\mathcal{S}$ and

$\mathcal{T}$ are invariant subspaces of a square matrix A of size m. Then, $\mathcal{S} = \operatorname{Im} S$ and $\mathcal{T} = \operatorname{Im} T$ and U defined by $U = [\ S \quad T\ ]$ satisfy

$$U^{-1}AU = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}.$$

If $\lambda(A_{11})$ ($\lambda(A_{22})$) lies in the closed right-half (open left-half) plane, then $\mathcal{S}$ ($\mathcal{T}$) is said to be the right (left) invariant subspace of A. When $\lambda(A_{11})$ ($\lambda(A_{22})$) lies outside (in) the open unit disk, then $\mathcal{S}$ ($\mathcal{T}$) is called the unstable (stable) invariant subspace of A. This notation is inherited from the stability of difference systems.

Let us now assume a regular matrix pencil $\lambda E - A$, which is a polynomial matrix (in the indeterminate λ) of degree one. The *generalized eigenvalue problem* for the matrices A and E of size m is equivalent to finding the scalars λ for which the equation $Ax = \lambda Ex$ has solutions $x \neq 0$. Such scalars λ are called generalized eigenvalues. A solution $x \neq 0$ corresponding to an eigenvalue λ is called a generalized eigenvector. A generalized eigenvalue satisfies the relation

$$\lambda \in \lambda(E, A) := \{\mu \in \mathbf{C} \mid \det(\mu E - A) = 0\},$$

where $\mathbf{C}$ is the field of complex numbers, and $\lambda(E, A)$ denotes the generalized spectrum of the matrix pair (E, A). Any subspace $\mathcal{S}$ satisfying

$$\mathcal{T} = E\mathcal{S} + A\mathcal{S},\ \dim(\mathcal{S}) = \dim(\mathcal{T})$$

is called a *generalized invariant subspace* (or a deflating subspace) of the pencil $\lambda E - A$. When $E = I$, we indeed have an ordinary invariant subspace.

Let $\mathcal{S}$ and $\mathcal{S}_c$ be two complementary deflating subspaces of the pencil $\lambda E - A$, i.e., $\mathcal{S} \oplus \mathcal{S}_c = \mathbf{R}^m$. Define $\mathcal{T} = E\mathcal{S} + A\mathcal{S}$ and $\mathcal{T}_c = E\mathcal{S}_c + A\mathcal{S}_c$. It is shown in [11] that these two subspaces are also complementary. Let $\mathcal{S} = \operatorname{Im} S$, $\mathcal{T} = \operatorname{Im} T$, $\mathcal{S}_c = \operatorname{Im} S_c$, and $\mathcal{T}_c = \operatorname{Im} T_c$. Then there exists a decomposition

$$U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix},\ U^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}$$

where

$$U = \begin{bmatrix} T & T_c \end{bmatrix},\ V = \begin{bmatrix} S & S_c \end{bmatrix}$$

If $\lambda(E_{11}, A_{11})$ ($\lambda(E_{22}, A_{22})$) lies in the closed right-half (open left-half) plane, then $\mathcal{S}$ ($\mathcal{S}_c$) is called the right (left) deflating subspace of the matrix pencil $\lambda E - A$. When $\lambda(E_{11}, A_{11})$ ($\lambda(E_{22}, A_{22})$) lies outside (in) the open unit disk, then $\mathcal{S}$ ($\mathcal{S}_c$) is called the unstable (stable) deflating subspace of the matrix pencil $\lambda E - A$.

2.4 Generalized State-space Approach

Now consider the Markov chain with the transition probability matrix given in (1). Define the d-transform of the sequence x_k, $k \geq 0$, as

$$x(d) \triangleq \sum_{k=0}^{\infty} x_k d^{-k}. \tag{20}$$

It is easy to show by (1) that

$$x(d)[I - dA(d)] = x_0[B(d) - dA(d)]. \tag{21}$$

Also define the sequence

$$y_k \triangleq x_{k+1},\ k \geq 0,$$

and let $y(d)$ be its d-transform. It is not difficult to show that

$$y(d) = d[x(d) - x_0] = x_0 N(d) D^{-1}(d), \tag{22}$$

where

$$\begin{aligned} N(d) &= d[R(d) - Q(d)] = N_1 d + N_2 d^2 + \cdots + N_f d^f, & (23)\\ D(d) &= Q(d) - dP(d) = D_0 + D_1 d + D_2 d^2 + \cdots + D_f d^f, & (24)\\ f &= q + 1. & (25)\end{aligned}$$

Here, f is called the *degree parameter* of the Markov chain and will play a key role in our approach.

Given the polynomial fraction (22), one can find a generalized state-space realization [12] for y_k (see Appendix B for a proof):

$$z_{k+1}E = z_k A,\ z_0 \bar{D} = x_0 \bar{N},\ \ y_k = z_k C,\ k \geq 0, \tag{26}$$

where

$$A = \begin{bmatrix} 0 & 0 & \cdots & \cdots & -D_0 \\ I & 0 & \cdots & \cdots & -D_1 \\ 0 & I & \cdots & \cdots & -D_2 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & I & -D_{f-1} \end{bmatrix},\ E = \begin{bmatrix} I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & D_f \end{bmatrix},\ C = \begin{bmatrix} I \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{27}$$

$$\bar{N} = \begin{bmatrix} N_f & N_{f-1} & N_{f-2} & \cdots & N_1 \end{bmatrix}, \tag{28}$$

and

$$\bar{D} = \begin{bmatrix} D_f & D_{f-1} & D_{f-2} & \cdots & D_1 \\ & D_f & D_{f-1} & \cdots & D_2 \\ & & D_f & \cdots & D_3 \\ & & & \ddots & \vdots \\ & & & & D_f \end{bmatrix}. \tag{29}$$

Here, z_k is is called the *descriptor* (or the semi-state) which reduces to the definition of *state* when E is nonsingular [21]. The possible singularity of E plays a significant role in the problem formulation. Also note that z_k is of size $1 \times mf$, and the matrices E and A are of size $mf \times mf$.

Remark: When the first ν coefficients of $D(d)$ are zero, i.e., $D_i = 0$ for $i = 0, 1, 2, \ldots, \nu - 1$, a reduced-order generalized state-space representation can be obtained. That is, the problem dimension can be reduced to

$m(f-\nu)$, resulting in an effective degree parameter of $f' = f - \nu$. In the case of a QBD chain, for example, it turns out that $D_0 = 0$. Therefore, the effective degree parameter can be made $f' = 2$ as opposed to $f = 3$ that (25) suggests. See [3] for details.

We now need to find x_0 so that none of the unstable modes of the matrix pair (E, A) are excited, i.e., z_k of (26) remains finite for all k. The matrix pencil $\lambda E - A$ has one singularity at $d = 1$, say, m_u singularities (including the one at $d = 1$) outside the open unit disk, and m_s singularities in the open unit disk. Note that $m_u + m_s$ yields the dimension mf of the generalized system given in (26). Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be the unstable and stable deflating subspaces of the pencil $\lambda E - A$, respectively. Let $\mathcal{V}_1 = \mathrm{Im}\, V_1$ and $\mathcal{V}_2 = \mathrm{Im}\, V_2$ for some matrices V_1 and V_2 of sizes $mf \times m_u$ and $mf \times m_s$, respectively. Also let $\mathcal{U}_1 := E\mathcal{V}_1 + A\mathcal{V}_1 = \mathrm{Im}\, U_1$ and $\mathcal{U}_2 := E\mathcal{V}_2 + A\mathcal{V}_2 = \mathrm{Im}\, U_2$ for some matrices U_1 and U_2 of sizes $mf \times m_u$ and $mf \times m_s$, respectively. Define

$$U \triangleq \begin{bmatrix} U_1 & U_2 \end{bmatrix} \quad \text{and} \quad V \triangleq \begin{bmatrix} V_1 & V_2 \end{bmatrix}. \tag{30}$$

Then, from Section 2.3, we have

$$U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix} \quad \text{and} \quad U^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \tag{31}$$

and $\lambda(E_{11}, A_{11})$ and $\lambda(E_{22}, A_{22})$ lie outside and in the open unit disk, respectively. Defining

$$\begin{bmatrix} u_k & v_k \end{bmatrix} \triangleq z_k \begin{bmatrix} U_1 & U_2 \end{bmatrix}, \ k \geq 0,$$

and post-multiplying the generalized state-space model (26) by V, we have two un-coupled generalized difference equations for u_k and v_k:

$$u_{k+1}E_{11} = u_k A_{11}, \ k \geq 0, \tag{32}$$

$$v_{k+1}E_{22} = v_k A_{22}, \ k \geq 0. \tag{33}$$

In order for z_k not to diverge as $k \to \infty$, u_0 must be the zero vector:

$$u_0 = z_0 U_1 = 0. \tag{34}$$

Moreover, since $\lambda(E_{22}, A_{22})$ lie in the open unit disk, E_{22} is nonsingular implying

$$v_k = v_0 F^k, \ k \geq 0, \tag{35}$$

where the $m_s \times m_s$ matrix F is found as

$$F = A_{22}E_{22}^{-1}. \tag{36}$$

Let us now partition U^{-1} as

$$U^{-1} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}, \tag{37}$$

where L_1 and L_2 has m_u and m_s rows, respectively. Then

$$x_{k+1} = y_k = z_k C = (u_k L_1 + v_k L_2)C = \underbrace{z_0 U_2}_{g} F^k \underbrace{L_2 C}_{H}, \quad k \geq 0. \tag{38}$$

The only unknowns that remain to complete the solution are x_0 and the initial value z_0 which, by (26) and (34), satisfy

$$\begin{bmatrix} x_0 & z_0 \end{bmatrix} Z = \begin{bmatrix} x_0 & z_0 \end{bmatrix} \begin{bmatrix} -\bar{N} & 0 \\ \bar{D} & U_1 \end{bmatrix} = 0. \tag{39}$$

Note that Z is $m(f+1) \times m(f+1)$. Furthermore, the sum of the probabilities x_k is unity which gives a normalizing equation in terms of x_0 and z_0:

$$\sum_{k=0}^{\infty} x_k e = x_0 e + \sum_{k=0}^{\infty} y_k e = x_0 e + z_0 U_2 (I - F)^{-1} H e = 1. \tag{40}$$

The concatenated vector $[x_0 \;\; z_0]$ is the unique solution to the two equations (39) and (40), which when computed leads to the simple matrix-geometric solution for the stationary probability vector x_k for level $k \geq 1$:

$$x_{k+1} = g F^k H, \; k \geq 0. \tag{41}$$

This simple and compact solution form makes it easier to write certain performance measures of interest. For example, the r-th factorial moment, $L(r)$, $r \geq 1$, of the level distribution is readily expressible in closed form as (also see [28])

$$L(r) = \sum_{k=r}^{\infty} \frac{k!}{(k-r)!} x_k e = r!\, g F^{r-1} (I - F)^{-1-r} H e. \tag{42}$$

The overflow probabilities, say P_B, are also easy to write:

$$P_B = \sum_{k=B+1}^{\infty} x_k e = g F^B (I - F)^{-1} H e. \tag{43}$$

In addition, the queue length distribution is known to exhibit a geometric decay as $\alpha\beta^k$ for sufficiently large k [34], [35], [10]. The form (41) of the solution indicates that the decay rate β here is the dominant eigenvalue of matrix F, which can be computed efficiently by the power method [17, Sec. 7.3.1]. More importantly, (41) allows computation of the coefficient α as well. Assuming that $\Lambda = T^{-1} F T$ is the Jordan form of F, the stationary probability of level $k+1$ can be written as $x_{k+1} e = g T \Lambda^k T^{-1} H e$. As k goes to infinity this expression reduces to $x_{k+1} e = g T_L \beta^k T_R H e$, where T_L and T_R are the left- and right-eigenvectors of F associated with the dominant eigenvalue β. Once β is computed, T_L and T_R can be found by solving two sets of linear equations, and then, the coefficient follows as $\alpha = g T_L T_R H e$.

This concludes the discussion of the existence of matrix-geometric solutions for infinite M/G/1 type Markov chains when the transform matrices $A(d)$ and $B(d)$ are rational functions of d. Two computational algorithms, one based on the matrix-sign function and the other on ordered generalized Schur decomposition, for finding the matrices U and V of decomposition (31) are presented in the next section.

3 Algorithms for Invariant Subspace Computations

The (generalized) invariant subspace computation (left or right, stable or unstable) is a well-known problem of numerical linear algebra [12], [16]. To name a few, (generalized) Schur decomposition methods [17], [20], inverse-free spectral divide and conquer methods [6], (generalized) matrix-sign function iterations [14] have been proposed to compute bases for these subspaces which arise for a wide variety of problems in applied mathematics. All of the above approaches can be used to find bases for the stable and unstable deflating subspaces of the matrix pencil $\lambda E - A$, which is an essential task in the generalized state-space method for solving M/G/1 type Markov chains. Here we present two algorithms: One is based on the *ordinary* matrix-sign function [32], and the other on the generalized Schur decomposition with ordering. The former algorithm employs certain properties of the matrices E and A akin to M/G/1 type models whereas the latter is quite generic. We also provide a summary of the overall method for infinite M/G/1 chains.

3.1 Matrix-sign Function Approach

We first note that, the stable (unstable) deflating subspace of the matrix pencil $\lambda E - A$ is equal to the left (right) deflating subspace of the pencil $\lambda L - M$, where the two matrices L and M are defined as

$$L \triangleq E + A, \quad M \triangleq A - E.$$

For a proof, we refer the reader to [14]. With this transformation, the generalized eigenvalues of the pencil $\lambda E - A$ in (outside) the open unit disk are moved to the open left-half (closed right-half) plane. So there is one generalized eigenvalue of $\lambda L - M$ on the imaginary axis, which is at the origin.

One can also show that L is regular by observing that the pencil $\lambda E - A$ does not have any generalized eigenvalue on the unit circle except one at $\lambda = 1$. In particular, $\lambda E - A$ does not have any generalized eigenvalue at $\lambda = -1$ which clearly shows that L is nonsingular. Let $W = L^{-1}M$. It is not difficult to show that the left (right) invariant subspace of W is also equal to the left (right) deflating subspace of the pencil $\lambda L - M$. Furthermore, W has one eigenvalue on the imaginary axis, which is at the origin. Then let γ and μ be left and right eigenvectors of W corresponding to the eigenvalue at the origin, i.e.,

$$\gamma W = 0, \quad W\mu = 0. \tag{44}$$

Then, the matrix W_e defined as

$$W_e \triangleq W + \frac{\mu\gamma}{\gamma\mu} \tag{45}$$

is free of imaginary-axis eigenvalues, and the left (right) invariant subspace of W_e is equal to the left (right) invariant subspace of W. It is not difficult to show that the vectors γ and μ defined as

$$\gamma \triangleq \begin{bmatrix} \pi & \pi & \cdots & \pi \end{bmatrix} L, \quad \mu \triangleq \begin{bmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_{f-1} \end{bmatrix}, \tag{46}$$

where π is the stationary probability vector of $A(1)$, i.e., $\pi A(1) = \pi$, $\pi e = 1$ and

$$\mu_0 = -D_0 Q^{-1}(1)e, \;\; \mu_i = \mu_{i-1} - D_i Q^{-1}(1)e, \; 1 \leq i \leq f-2, \;\; \mu_{f-1} = Q^{-1}(1)e,$$

satisfy (44).

One can now use matrix-sign function iterations on W_e [2] to find bases for the unstable and stable deflating subspaces of the pencil $\lambda E - A$, leading to construction of the matrices U and V defined as in (30). We now outline this approach.

We refer the reader to [2] for the definition of the matrix-sign function. The basic matrix-sign function algorithm for an $m \times m$ matrix M with no imaginary axis eigenvalues is (see [7] and [14])

$$Z_0 = M, \; Z_{k+1} = \frac{1}{2c_k}(Z_k + c_k^2 Z_k^{-1}), \; c_k = |\det(Z_k)|^{1/m}. \tag{47}$$

Then,

$$\lim_{k\to\infty} Z_k = Z = sgn(M),$$

where $sgn(M)$ denotes the matrix sign of M, and convergence is quadratic. The stopping criterion we use is the one proposed in [5]:

$$||Z_{k+1} - Z_k||_1 < \epsilon ||Z_k||_1. \tag{48}$$

The most important property of matrix sign is that $\mathrm{Im}\,(Z - I)$ $(\mathrm{Im}\,(Z + I))$ is equal to the left (right) invariant subspace of M [32]. Then find

$$S = sgn(W_e) \tag{49}$$

through the matrix-sign function iterations (47). Recall that there are m_u eigenvalues of W_e in the right-half plane and m_s eigenvalues in the left-half plane. Let the rank-revealing QR decomposition [8] of $S + I$ be

$$S + I = Q_r R_r \Pi_r, \tag{50}$$

where R_r is upper triangular, Q_r is orthogonal, and Π_r is a permutation matrix. Suppose that Π_r is chosen so that the rank deficiency of $S+I$ is exhibited in R_r by a smaller lower-right block in norm of size $m_s \times m_s$. Then, let

$$V_1 \triangleq \text{ leading } m_u \text{ columns of } Q_r, \tag{51}$$

which span $\mathrm{Im}\,(S+I)$, or equivalently, form an orthogonal basis for the left-invariant subspace of W_e or the unstable deflating subspace of the pencil $\lambda E - A$. Similarly, a rank-revealing QR decomposition of $S - I$ yields

$$S - I = Q_l R_l \Pi_l, \tag{52}$$

and with a proper choice of permutation Π_l, we define

$$V_2 \triangleq \text{leading } m_s \text{ columns of } Q_l. \tag{53}$$

Following Section 2.3, with two more rank-revealing QR decompositions,

$$\begin{aligned} \left[\begin{array}{cc} EV_1 & AV_1 \end{array}\right] &= \hat{Q}_r \hat{R}_r \hat{\Pi}_r, && (54) \\ \left[\begin{array}{cc} EV_2 & AV_2 \end{array}\right] &= \hat{Q}_l \hat{R}_l \hat{\Pi}_l, && (55) \end{aligned}$$

we define

$$\begin{aligned} U_1 &\triangleq \text{leading } m_u \text{ columns of } \hat{Q}_r, && (56) \\ U_2 &\triangleq \text{leading } m_s \text{ columns of } \hat{Q}_l. && (57) \end{aligned}$$

This concludes the discussion of using ordinary matrix-sign function to find the four key matrices V_1, V_2, U_1, and U_2 that are used to decompose the generalized system (26) into its forward and backward subsystems through (31).

3.2 Generalized Schur Decomposition Approach

One other approach to find the stable (unstable) deflating subspaces that give rise to decomposition (31) is to use generalized Schur decomposition with ordering [17], [20]. Given the two matrices E and A, one can employ the generalized Schur decomposition method with ordering [20] to compute the two orthonormal matrices Θ and Ψ satisfying

$$\Theta^T E \Psi = \left[\begin{array}{cc} E_{11} & E_{12} \\ 0 & E_{22} \end{array}\right], \quad \Theta^T A \Psi = \left[\begin{array}{cc} A_{11} & A_{12} \\ 0 & A_{22} \end{array}\right], \tag{58}$$

such that

1. A_{11} and A_{22} are upper triangular with non-negative diagonals,
2. E_{11} and E_{22} are upper block triangular with either 1-by-1 or 2-by-2 blocks (corresponding to complex eigenvalues)
3. $\lambda(E_{11}, A_{11})$ lies in the open unit disk,
4. $\lambda(E_{22}, A_{22})$ lies outside the open unit disk.

Given the above decomposition, we next solve the generalized Sylvester equations [20]

$$A_{11}Y - XA_{22} = A_{12} \quad \text{and} \quad E_{11}Y - XE_{22} = E_{12} \tag{59}$$

for the two matrices X and Y. Finally, defining

$$U \triangleq \Theta \left[\begin{array}{cc} I & -X \\ 0 & I \end{array}\right] \quad \text{and} \quad V \triangleq \Psi \left[\begin{array}{cc} I & -Y \\ 0 & I \end{array}\right], \tag{60}$$

one obtains the decomposition (31), i.e., eliminates the upper-diagonal blocks E_{12} and A_{12} in (58). Here we note that elimination of these blocks is not necessary in the solution of infinite M/G/1 chains. However, for the case of finite M/G/1 chains that will be discussed in the next section, those blocks have to be eliminated.

In Table 1, we provide an algorithmic description of the generalized state-space approach for infinite M/G/1 chains based on either the matrix-sign function or the generalized Schur decomposition with ordering. The algorithm assumes that the right polynomial fractions of (12) are given.

4 Finite M/G/1 Type Markov Chains

Consider the finite M/G/1 chain of the form (3). It is not difficult to show that the generalized difference equations (26) are still valid for $0 \leq k \leq K-1$:

$$z_{k+1}E = z_k A, \; z_0 \bar{D} = x_0 \bar{N}, \quad y_k = z_k C, \quad 0 \leq k \leq K-1. \tag{61}$$

Using the same decomposition (31) as in infinite M/G/1 chains, we have

$$u_{k+1}E_{11} = u_k A_{11}, \; 0 \leq k < K-1, \tag{62}$$

or, equivalently

$$u_k = u_{k+1}F_2, \quad 0 \leq k < K-1 \tag{63}$$

where

$$F_2 = E_{11}A_{11}^{-1}. \tag{64}$$

The invertibility of A_{11} follows directly from the fact that the generalized eigenvalues of the pair (E_{11}, A_{11}) lie outside the open unit disk. Therefore, the matrix F_2 has all its eigenvalues in the closed unit disk. We call (63) the *backward subsystem* of the generalized system (61). It immediately follows from (63) that

$$u_k = u_{K-1}F_2^{K-k-1}, \; 0 \leq k < K-1. \tag{65}$$

The main difference from the infinite M/G/1 formulation is that, the unstable modes of the pair (E, A) may be excited in finite M/G/1 chains and the vector u_0 is not necessarily the zero vector. On the other hand, the difference equations corresponding to the *forward subsystem* (33) are still valid for $0 \leq k \leq K-1$, leading to

$$v_k = v_0 F_1^k, \; 0 \leq k \leq K-1, \tag{66}$$

where $F_1 \triangleq F$. Then, the solution y_k, $0 \leq k \leq K-1$, of the finite M/G/1 chain of (3) can be written in terms of u_{K-1} and v_0:

$$y_k = z_k C = \underbrace{v_0}_{g_1} F_1^k \underbrace{L_2 C}_{H_1} + \underbrace{u_{K-1}}_{g_2} F_2^{K-k-1} \underbrace{L_1 C}_{H_2}, \; 0 \leq k \leq K-1, \tag{67}$$

proving the existence of the modified matrix-geometric form given in (8). What now remains is to find the three unknown vectors x_0, g_1, and g_2. We have got two equations to solve for these vectors, the first of which is $z_0\bar{D} = x_0\bar{N}$ yielding

$$g_1 L_2 \bar{D} + g_2 F_2^{K-1} L_1 \bar{D} = x_0 \bar{N} \tag{68}$$

through straightforward substitution. The second equation derived from the balance equation at level K is

$$x_0 \bar{B}_{K-1} + \sum_{i=0}^{K-1} y_i \bar{A}_{K-i-1} = y_{K-1},$$

and can be rewritten in terms of x_0, g_1, and g_2 by using (67) as

$$x_0 \bar{B}_{K-1} + g_1 \underbrace{\left(\sum_{i=0}^{K-1} F_1^i H_1 \bar{A}_{K-i-1} \right)}_{\Delta_1} + g_2 \underbrace{\left(\sum_{i=0}^{K-1} F_2^{K-1-i} H_2 \bar{A}_{K-i-1} \right)}_{\Delta_2} = g_1 F_1^{K-1} H_1 + g_2 H_2. \tag{69}$$

Using (68) and (69), one can then find x_0, g_1, and g_2 uniquely by solving the equation

$$\begin{bmatrix} x_0 & g_1 & g_2 \end{bmatrix} Z_e = \begin{bmatrix} x_0 & g_1 & g_2 \end{bmatrix} \begin{bmatrix} -\bar{N} & \bar{B}_{K-1} \\ L_2 \bar{D} & \Delta_1 - F_1^{K-1} H_1 \\ F_2^{K-1} L_1 \bar{D} & \Delta_2 - H_2 \end{bmatrix} = 0, \tag{70}$$

and normalizing the solution such that the stationary probabilities add up to unity, i.e., $\sum_{i=0}^{K} x_k e = 1$.

5 M/G/1 Chains with Multiple Boundaries

Based on [3], we outline below the algorithm for finding the matrix-geometric factors of the M/G/1 chain with multiple boundary levels. The proof is similar to that of the canonical M/G/1 chain and is omitted in this paper.

Consider the M/G/1 type Markov chain in (9) with N multiple boundary levels. First define

$$A(d) \triangleq \sum_{j=0}^{\infty} A_j d^{-j} \quad \text{and} \quad B_i(d) \triangleq \sum_{j=0}^{\infty} B_{i,j} d^{-j},\ 0 \le i < N. \tag{71}$$

Then find a stable right coprime fraction as

$$\begin{bmatrix} A(d) \\ B_0(d) \\ B_1(d) \\ \vdots \\ B_{N-1}(d) \end{bmatrix} = \begin{bmatrix} P(d) \\ R_0(d) \\ R_1(d) \\ \vdots \\ R_{N-1}(d) \end{bmatrix} Q^{-1}(d). \tag{72}$$

Let q be the degree of the polynomial matrix $Q(d)$ and define

$$D(d) = D_0 + D_1 d + D_2 d^2 + \cdots + D_f d^f = Q(d) - d^N P(d), \tag{73}$$

and

$$N(d) = N_1 d + N_2 d^2 + \cdots + N_f d^f = \begin{bmatrix} R_0(d)d^N - d^N Q(d) \\ R_1(d)d^N - d^{N-1}Q(d) \\ \vdots \\ R_{N-1}(d)d^N - dQ(d) \end{bmatrix}. \tag{74}$$

Note that $N(d)$ and $D(d)$ are polynomial matrices of degree $f = q + N$. Define the matrices E, A, C, $\bar{N}$ and $\bar{D}$ in the same way as in (27),(28), and (29) using the polynomials of (73) and (74) above. The rest of the algorithm is the same as that for the canonical M/G/1 chain. We first find the matrices U, V, and F as in (30) and (36), and partition U^{-1} as in (37). We then solve for

$$\begin{bmatrix} x_0 & x_1 & \cdots & x_{N-1} & z_0 \end{bmatrix} \begin{bmatrix} -\bar{N} & 0 \\ \bar{D} & U_1 \end{bmatrix} = 0, \tag{75}$$

and normalize the solution so that

$$\sum_{i=0}^{N-1} x_i e + z_0 U_2 (I - F)^{-1} H e = 1. \tag{76}$$

Defining $g \triangleq z_0 U_2$ and $H \triangleq L_2 C$ gives us the matrix-geometric solution

$$x_{k+N} = g F^k H, \; k \geq 0. \tag{77}$$

6 Numerical Examples and Discussion

Example 1. We first consider an infinite M/G/1 type Markov chain obtained from the $\text{IPP}_n/\text{E}_r/1$ queueing model, where IPP_n stands for the superposition of n independent and identical IPP's (Interrupted Poisson Process) [13] and E_r stands for the r-stage Erlangian distribution. We construct and refer to this chain through the following three parameters: *(i)* the number of phases m, *(ii)* the degree parameter f defined by (25), and *(iii)* the traffic parameter (or utilization) ρ given in (17). Note that, since i.i.d. IPP's are considered, setting $n = m - 1$ results in an m-state Markovian model for the aggregate arrival process. For each IPP source, we fix the transition rates to the idle and active states as 3 and 1, respectively. Therefore, the arrival rate in the active state of each IPP is uniquely determined for any desired aggregate arrival rate λ. We fix the mean service rate as $\mu = 1$, which implies that $\rho = \lambda$. Since $\text{IPP}_n/\text{E}_r/1$ is a special case of MMPP/G/1 queueing model, the probability generating matrices $A(d)$ and $B(d)$ are found as described in Section 2.1. The Laplace-Stieltjes transform of an r-stage Erlangian distribution with unity mean is given as [22] $h(s) = (1 + s/r)^{-r}$. Hence, a desired degree parameter f is met by setting $r = f - 1$.

We provide CPU time and error results for the Matrix-Sign Function (MSF), Generalized Schur Decomposition (GSD) implementations of the generalized state-space approach (see Sections 2.4 and 3), and the truncation-free Successive Substitution Iteration (SSI) method outlined in Section 2.2. We measure the CPU time until the point at which the program becomes ready to compute the level probability vectors x_k, $k > 0$. This amounts to finding the matrix-geometric factors g, F, and H in the case of the MSF and GSD methods, and to finding the level-0 probability vector x_0 and the matrix sequences $\{\hat{A}_k\}$ and $\{\hat{B}_k\}$ in the case of SSI method. As for error computations, we consider the infinity norm of the residual of the solution vector: $||x - xP||_\infty$; see (1) and (2). However, since x and P are infinite entities, truncation is needed here. We simply consider the balance equation over levels 0 and 1 only. That is, we estimate the error as $||\bar{x} - \bar{x}\bar{P}||_\infty$, where $\bar{x} = [\ x_0 \quad x_1 \quad x_2\ ]$ and $\bar{P}$ is the corresponding partition of P. Note that all three matrix-geometric factors of the solution are involved in this computation.

All MSF, GSD, and SSI methods are implemented in C, and compiled (by gcc version 2.7.2.1 with optimizer `-O3`) and run on a DEC Alpha server supporting IEEE standard double-precision arithmetic with machine epsilon $\varepsilon \approx 2.2e - 16$. Standard BLAS (Basic Linear Algebra Subroutines) and CLAPACK (Fortran-to-C translated version of LAPACK–Linear Algebra PACKage) library routines [4] are used to perform all matrix operations[2]. In the SSI implementation, all polynomial matrix evaluations are performed using Horner's method, and the $\{\hat{A}_k\}$ and $\{\hat{B}_k\}$ matrix sequences are obtained by backward recursions on the $\{A_k\}$ and $\{B_k\}$ matrix sequences which are truncated at indices K_1 and K_2 such that $\sum_{k>K_1} A_k e$ and $\sum_{k>K_2} B_k e$ have negligible components [24]. In all iterations, $\epsilon = 10^{-8}$ is used in the stopping criteria. In the SSI method, since the G_* matrix is known to be stochastic, we use $||G_* e - e||_\infty < \epsilon$ as the stopping criterion.

In Table 2, fixing the utilization as $\rho = 0.6$, we present the CPU time and error results for different values of m and f. These results show that the MSF and GSD methods are highly accurate although they may incur more computational complexity than the SSI method as f increases. It is noteworthy that the GSD method outperforms the (serial implementation) of the MSF method in terms of CPU time. Since $f = 4$ is the largest degree parameter considered here, we evaluate polynomials of degree at most three during the truncation-free successive substitution iteration of (15). The number of A_k matrices generated ($K_1 \approx$ 14-21) indicates the significant time savings achieved here by exploiting the rational structure in (15) in comparison to the traditional iteration [29]

$$G_0 = 0, \quad G_{j+1} = \sum_{k=0}^{K_1} A_k G_j^k.$$

In Table 3, we provide the same set of results as in Table 2 for utilization $\rho = 0.9$. As these results indicate, the MSF and GSD methods are still very accurate. Furthermore, unlike the case for matrix-sign function iterations, the number of successive substitution iterations increase substantially with utilization, and the MSF method becomes faster than the SSI method as utilization increases unless the degree parameter f is large. To further explore this point, we fix m and f as 32 and 3, respectively, and perform a stress test with

[2] Here we note that the routines for generalized Schur decomposition with ordering and generalized Sylvester equation solution, which are required to implement the GSD method are not available in the current version of LAPACK. However, they will become available in the new release (version 3.0) of the package. The respective routines that we have obtained via private communication and used in our implementations are DGGES and DTGSYL.

respect to utilization. The results shown in Table 4 indicate the efficiency and high numerical stability of the MSF and GSD methods under heavy load conditions which yield a very ill-conditioned numerical problem. In fact, Schur decomposition is in general known of its remarkable numerical stability [6]. This is observed as the GSD method becomes more accurate than the MSF method with increasing ρ. The fact that the CPU time for the GSD method is not affected at all by utilization is particularly noteworthy here.

Note that, in Table 4, the error of the SSI method decreases as utilization is increased. This counter-intuitive trend may be due to the fact that the probability mass moves to higher levels with increasing utilization, which makes error measurements through the balance equations over only levels 0 and 1 less representative of the actual error.

Example 2. We now consider a finite M/G/1 type Markov chain of the form (3) with $A_k = 0$ for $k \geq f$. The way $A(d)$ is obtained in this case was discussed in Section 2.1. We note that the effective degree parameter in this example can be made less than f (see the remark in Section 2.4); however, we did not exploit this possibility in our implementation.

The $m \times m$ matrices A_k, $0 \leq k < f$, are specified as follows: *(i)* A_0 and A_{f-1} are both diagonal matrices with constant diagonal entries $1 - 2r$ and r, respectively, where $0 < r < 0.5$; *(ii)* All other A_k, $0 < k < f - 1$, are equal to each other, and tridiagonal with null diagonal entries, i.e., their nonzero entries are $a_{i,i+1} = a_{i,i-1} = 0.5r/(f-2)$, $0 < i < m - 1$, and $a_{0,1} = a_{m-1,m-2} = r/(f-2)$. It can be shown that the utilization of this system is $\rho = 3(f-1)r/2$, irrespective of m. So, given f and ρ, the parameter r is uniquely determined, and the number of phases m can be arbitrarily chosen.

Having defined A_k's, we assume the level-0 boundary behavior of (5), and B_k's follow accordingly. As for the limiting level K, we use (4) to obtain $\bar{B}_{K-1}$ and $\bar{A}_k$'s. The hardware and software platform used is the same as described in Example 1.

Table 5 presents the CPU time and error results obtained using the generalized Schur decomposition method (see Sections 2.4 and 3.2). As was observed in Example 1, the CPU time for this method is insensitive to utilization. Therefore, in Table 5, we only provide CPU time versus K results, and this time, the measurements are taken both after finding the matrix-geometric factors of the solution form and after all $K + 1$ level probability vectors are computed. We also provide the CPU times for the infinite-level counterpart of the original chain. Note that the effect of the number of levels on the CPU time is minimal in this example. In addition, even for K as high as 1000, the solution of the finite chain takes about twice the time required to solve the infinite chain. We are thus led to believe that approximating finite M/G/1 chains by their infinite counterparts may be unnecessary in many circumstances. On the other hand, the error is computed as the infinity norm of the residual of the complete solution vector: $||x - xP||_\infty$; see (3) and (2). Apart from a reasonable worsening for utilization very close to unity, Table 5 verifies the accuracy and numerical stability of our method under this ultimate error measure.

Example 3. This example is on an infinite M/G/1 chain with multiple boundary levels arising in queueing systems with multiple servers. We assume a discrete-time, slotted queueing system with the following evolution equation for the queue length:

$$Q_{n+1} = \max(0, Q_n + C_n - N) + r_n, \; n \geq 0,$$

where Q_n is the queue length at the end of the n-th slot, $N \geq 1$ is the number of servers, C_n is the total number of arrivals of type 1 traffic in the n-th slot with $C_n \leq N$, r_n is the total number of arrivals in the n-th slot due to type 2 traffic in the n-th slot, and type 2 traffic is buffered when the servers are busy. C_n is modulated by a homogeneous finite-state, aperiodic discrete-time Markov chain with state transitions taking place only at the slot boundaries. Let S_n be the state of the modulating chain before the end of slot n. We also have

$$c_{ij}(d) = E[d^{-C_n}\mathbf{1}(S_{n+1} = j | S_n = i)], \; 1 \leq i,j \leq m, \;\; C(d) = \{c_{ij}(d)\},$$

where $\mathbf{1}(E)$ is equal to one if the event E is true and zero otherwise. Note that $C(d)$ takes the following form

$$C(d) = \sum_{i=0}^{M} C_i d^{-i}, \; M \leq N.$$

We also assume r_n to be an independent geometric batch process with parameter a, i.e.,

$$r(d) = E[d^{-r_n}] = \frac{(1-a)d}{d-a}.$$

This queueing system is suitable for modeling voice and data traffic multiplexing over a single channel, and falls into the M/G/1 paradigm with multiple boundary levels with the doublet (Q_n, S_n) being Markov and having a transition probability matrix of the form (9). It is not difficult to show that $B_i(d)$, $0 \leq i \leq N-1$, and $A(d)$ can now be written as

$$\begin{aligned}
B_0(d) &= (C_0 + C_1 + \cdots + C_M) r(d) \\
B_1(d) &= (C_0 + C_1 + \cdots + C_M) r(d) \\
&\vdots \\
B_{N-M+1}(d) &= (C_M d^{-1} + (C_0 + C_1 + \cdots + C_{M-1})) r(d) \\
B_{N-M+2}(d) &= (C_M d^{-2} + C_{M-1} d^{-1} + (C_0 + C_1 + \cdots + C_{M-2})) r(d) \\
&\vdots \\
A(d) &= (C_M d^{-M} + C_{M-1} d^{-M+1} + \cdots + C_0) r(d)
\end{aligned}$$

One can then obtain a stable matrix fraction as in (72) by choosing

$$Q(d) = \frac{1}{1-a} d^{N-1}(d-a) I$$

and

$$R_0(d) = (C_0 + C_1 + \cdots + C_M) d^N$$

$$
\begin{aligned}
R_1(d) &= (C_0 + C_1 + \cdots + C_M) d^N \\
&\vdots \\
R_{N-M+1}(d) &= C_M d^{N-1} + (C_0 + C_1 + \cdots + C_{M-1}) d^N \\
R_{N-M+2}(d) &= C_M d^{N-2} + C_{M-1} d^{N-1} + (C_0 + C_1 + \cdots + C_{M-2}) d^N \\
&\vdots \\
P(d) &= C_M d^{N-M} + C_{M-1} d^{N-M+1} + \cdots + C_0 d^N
\end{aligned}
$$

With the polynomial matrix fractions obtained as above, we implemented the method outlined in Section 5 using MATLAB on the same hardware platform as in Example 1. We used the matrix-sign function approach of Section 3.1. We take $N = 2$, $M = 2$, and

$$
C(d) = \underbrace{\begin{bmatrix} \text{9.9996e-1} & 0 \\ \text{1.0000e-5} & 0 \end{bmatrix}}_{C_0} + \underbrace{\begin{bmatrix} 0 & \text{3.0000e-5} \\ 0 & \text{7.4999e-1} \end{bmatrix}}_{C_1} d^{-1} + \underbrace{\begin{bmatrix} 0 & \text{1.0000e-5} \\ 0 & \text{2.5000e-1} \end{bmatrix}}_{C_2} d^{-2}
$$

which amounts to a bursty traffic of type 1 yielding a load of %50 on the system. The traffic parameter a is chosen so as to yield a desired overall arrival rate. We vary the total load ρ on the system and present the results in Table 6. The error is computed as the infinity norm of the difference between the left- and right-hand sides of equation (75). $P[empty] = x_0 e$ is the probability of empty queue, and $E[queue]$ is the expected queue length. To give an example about the detailed results, we have obtained the following matrices that constitute the matrix-geometric form (10) for $\rho = 0.99$:

$$
g = \begin{bmatrix} -\text{1.076398e-3} & \text{2.319338e-4} \end{bmatrix}, \; F = \begin{bmatrix} \text{5.408260e-1} & \text{9.895524e-2} \\ -\text{3.165397e-1} & \text{1.068213e-0} \end{bmatrix},
$$

$$
H = \begin{bmatrix} -\text{6.066254e0} & -\text{1.852386e1} \\ -\text{2.068470e1} & -\text{8.595458e1} \end{bmatrix}.
$$

We note that these matrices do not have a probabilistic interpretation unlike the case for matrix-analytic approach for M/G/1 and G/M/1 type Markov chains [28], [29]. We have also found the geometric decay rate for the level distribution, i.e., the eigenvalue of F closest to the unit circle, which happens to be $\beta = 0.999996$ for this example.

Certain advantages of the simple and compact matrix-geometric form for the stationary solution of M/G/1 chains were addressed briefly at the end of Section 2.4. The results of this section demonstrate the accuracy and numerical stability of two particular implementations (based on ordinary matrix-sign function and generalized Schur decomposition with ordering) of the generalized state-space approach of this paper. The results also indicate substantial savings in the CPU time and storage requirements compared to conventional recursive techniques especially when the degree parameter f is not large. However, in the case of intolerably large f, rational approximation techniques can be employed to reduce f and decrease the computational complexity with insignificant loss of accuracy. For example, in [1], the deterministic service time in a MAP/D/1/K queueing system, which indeed results in $f = \infty$, is modeled by Padé approximations in transform domain with a reduced degree of $f = 3$, and very accurate estimates for cell loss rates in an ATM multiplexer are obtained efficiently.

A Polynomial Matrices and Fractions

The following material on polynomial matrices and polynomial matrix fractions of a rational matrix $A(d)$ is mainly based on [21] and [9].

A matrix $A(d) = [a_{ij}(d)]$, where $a_{ij}(d) = p_{ij}(d)/q_{ij}(d)$ for a polynomial pair $p_{ij}(d)$ and $q_{ij}(d)$, is called a rational matrix in d. If $\deg(p_{ij}) \le \deg(q_{ij})$ ($\deg(p_{ij}) < \deg(q_{ij})$) for all i and j, then $A(d)$ is called *proper* (*strictly proper*). We say $A(d)$ is *stable* if all roots of $q_{ij}(d)$ lie in the open unit disk for all i and j. A polynomial matrix is one with polynomial entries. Let $Q(d)$ and $P(d)$ be $m_q \times m_q$ and $m_p \times m_q$ polynomial matrices, respectively, and let $Q(d)$ be nonsingular. $Q(d)$ and $P(d)$ are said to be *right coprime* over an arbitrary region $\mathbf{D}$ in the complex plane $\mathbf{C}$ if and only if, for every $d \in \mathbf{D}$, the matrix $[\ Q^T(d) \quad P^T(d)\ ]^T$ has full column rank m_q. If in the above definition $\mathbf{D}$ is outside the open unit disk, that is, if $\mathbf{D}$ is the set $\{d \in \mathbf{C},\ |d| \ge 1\}$, then the fraction is called *stable* right coprime. Consider a rational matrix $A(d)$ of size $m_p \times m_q$. The fraction $A(d) = P(d)Q^{-1}(d)$ is called a stable right coprime polynomial fraction if the polynomial matrices $Q(d)$ and $P(d)$ are stable right coprime. Given a stable rational matrix $A(d)$, it is always possible to obtain a stable right coprime polynomial fraction [9].

B Generalized State-space Realization

Here we provide a proof for the generalized state-space realization (26) for y_k. Following the treatment of [1] in a similar context, we define

$$y_k^j \triangleq y_{k+j-1}, \quad y^j(d) = \sum_{k=0}^{\infty} y_k^j d^{-k}, \quad 1 \le j \le f,$$

which yields

$$\begin{aligned} y^1(d) &= y(d), \\ y^j(d) &= d y^{j-1}(d) - d y_0^{j-1},\ 2 \le j \le f. \end{aligned}$$

Here, j in y_k^j should be treated as a superscript. We note that

$$y_0^j = \lim_{d \to \infty} y^j(d), \quad 1 \le j \le f,$$

must be a bounded vector by definition. It is also easy to see that

$$d y^j(d) = y^{j+1}(d) + d y_0^{j-1}, \quad 1 \le j \le f-1. \tag{78}$$

One can also show by using (22) and by algebraic manipulations that

$$dy^f(d)D_f = -\sum_{i=0}^{f-1} y^{i+1}(d)D_i + \sum_{i=1}^{f} x_0 N_i d^i - \sum_{i=1}^{f-1}\sum_{r=1}^{i} d^{i+1-r} y_0^r D_i - d\sum_{i=1}^{f-1} d^{f-i} y_0^i D_f. \tag{79}$$

Since $\lim_{d\to\infty} y^f(d)D_f$ exists, (79) dictates linear constraints on y_0^j, $1 \le j \le f$, in the following manner:

$$y_0^j D_f = x_0 N_{f-j+1} - \sum_{i=1}^{j-1} y_0^i D_{f-j+i}. \tag{80}$$

Consequently, (79) becomes

$$dy^f(d)D_f = -\sum_{i=0}^{f-1} y^{i+1}(d)D_i + dy_0^f D_f. \tag{81}$$

Let us now define the concatenated transform vector

$$z(d) \triangleq \begin{bmatrix} y^1(d) & y^2(d) & \cdots & y^f(d) \end{bmatrix},$$

which is the d-transform of the sequence

$$z_k = \begin{bmatrix} y_k^1 & y_k^2 & \cdots & y_k^f \end{bmatrix}.$$

One can then make use of (78), (80), and (81) to obtain

$$z(d)(dE - A) = dz_0, \quad z_0\bar{D} = x_0\bar{N}, \quad \text{and} \quad y(d) = z(d)C, \tag{82}$$

where the matrices A, E, C, $\bar{D}$, and $\bar{N}$ are as defined in (27),(28), and (29), respectively.

We note that the transform identities in (82) are equivalent to the representation (26) and therefore we have found one particular generalized state-space realization for the transform expression (22) for $y(d)$.

References

[1] N. Akar and E. Arıkan. A numerically efficient algorithm for the MAP/D/1/K queue via rational approximations. *Queueing Systems: Theory and Applications*, 22:97–120, 1996.

[2] N. Akar and K. Sohraby. An invariant subspace approach in M/G/1 and G/M/1 type Markov chains. *Stochastic Models*, 13(3), 1997.

[3] N. Akar and K. Sohraby. Matrix-geometric solutions in M/G/1 type Markov chains with multiple boundaries: a generalized state-space approach. In *Proc. ITC'15*, pages 487–496, 1997.

[4] E. Anderson *et al.* *LAPACK Users's Guide, 2-nd ed.*, 1994. Also available at the URL http://www.netlib.org.

[5] Z. Bai and J. Demmel. Design of a parallel nonsymmetric eigenroutine toolbox: Part I. Computer Science Division Report CSD-92-718, University of California at Berkeley, Dec. 1992.

[6] Z. Bai, J. Demmel, and M. Gu. Inverse free parallel spectral divide and conquer algorithms for nonsymmetric eigenproblems. Computer Science Division Report CSD-94-793, University of California at Berkeley, Feb. 1994.

[7] R. Byers. Solving the algebraic Riccati equation with the matrix sign function. *Lin. Alg. Appl.*, 85:267–279, 1987.

[8] T. F. Chan. Rank revealing QR factorizations. *Lin. Alg. Appl.*, 88/89:67–82, 1987.

[9] C. T. Chen. *Linear System Theory and Design.* Holt, Rinehart and Winston, Inc., New York, 1984.

[10] G. L. Choudhury, D. M. Lucantoni, and W. Whitt. Squeezing the most out of ATM. *IEEE Trans. Commun.*, 44:203–217, 1996.

[11] J. Demmel and B. Kagström. Computing stable eigendecompositions of matrix pencils. *Lin. Alg. Appl.*, 88/89:139–186, 1987.

[12] P. M. Van Dooren. The generalized eigenstructure problem in linear system theory. *IEEE Trans. Automat. Contr.*, 26(1):111–129, 1981.

[13] W. Fischer and K. Meier-Hellstern. The Markov-modulated Poisson process (MMPP) cookbook. *Performance Evaluation*, 18:149–171, 1992.

[14] J. D. Gardiner and A. J. Laub. A generalization of the matrix-sign-function solution for algebraic Riccati equations. *Int. Jour. Contr.*, 44:823–832, 1986.

[15] J. D. Gardiner and A. J. Laub. Parallel algorithms for the algebraic Riccati equation. *International Jour. Contr.*, 54(6):1317–1333, 1991.

[16] I. C. Gohberg, P. Lancaster, and L. Rodman. *Invariant Subspaces of Matrices with Applications.* John Wiley and Sons, New York, 1986.

[17] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 1989.

[18] H. Heffes and D. M. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE JSAC*, 4(6):856–868, 1986.

[19] C. L. Hwang and S. Q. Li. On the convergence of traffic measurement and queueing analysis: A statistical-match queueing (SMAQ) tool. In *Proc. IEEE INFOCOM*, pages 602–612, 1995.

[20] B. Kagström. A direct method for reordering eigenvalues in the generalized Schur form of a regular matrix pair (A,B). In M. S. Moonen et al., editor, *Linear Algebra for Large Scale and Real-time Applications*, pages 195–218. Kluwer Academic Publ., 1993.

[21] T. Kailath. *Linear Systems.* Prentice-Hall Inc., Englewood Cliffs NJ, 1980.

[22] L. Kleinrock. *Queueing Systems Volume 1: Theory.* Wiley-Interscience Publication, 1975.

[23] G. Latouche. A note on two matrices occurring in the solution of quasi-birth-and-death processes. *Commun. Statist. - Stochastic Models*, 3(2):251–257, 1987.

[24] D. M. Lucantoni. New results for the single server queue with a batch Markovian arrival process. *Stoch. Mod.*, 7:1–46, 1991.

[25] D. M. Lucantoni. The BMAP/G/1 queue: A tutorial. In L. Donatiello and R. Nelson, editors, *Models and Techniques for Performance Evaluation of Computer and Communication Systems*, pages 330–358. Springer-Verlag, 1993.

[26] D. M. Lucantoni, K. S. Meier-Hellstern, and M. F. Neuts. A single-server queue with server vacations and a class of non-renewal arrival processes. *Adv. Appl. Prob.*, 22:676–705, 1990.

[27] B. Melamed, Q. Ren, and B. Sengupta. Modeling and analysis of a single server queue with autocorrelated traffic. In *Proc. IEEE INFOCOM*, pages 634–642, 1995.

[28] M. F. Neuts. *Matrix-geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore, MD, 1981.

[29] M. F. Neuts. *Structured Stochastic Matrices of M/G/1 Type and Their Applications.* Marcel Dekker, Inc., New York, 1989.

[30] M. F. Neuts. Matrix-analytic methods in queuing theory. In J. H. Dshalalow, editor, *Advances in Queueing: Theory, Methods, and Open Problems*, pages 265–292. CRC Press IUnc.,, 1995.

[31] V. Ramaswami. A stable recursion for the steady-state vector in markov chains of M/G/1 type. *Commun. Statist.-Stoc. Models*, 4:183–188, 1988.

[32] J. D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by the use of the sign function. *Int. Jour. Control*, 32:677–687, 1980.

[33] H. Schellhaas. On Ramaswami's algorithm for the computation of the steady-state vector in Markov chains of M/G/1 type. *Commun. Statist.-Stoc. Models*, 6(3):541–550, 1990.

[34] K. Sohraby. On the asymptotic behavior of heterogeneous statistical multiplexer with applications. In *Proc. IEEE INFOCOM*, pages 839–847, 1992.

[35] K. Sohraby. On the theory of general on-off sources with applications in high-speed networks. In *Proc. IEEE INFOCOM*, 1993.

[36] V. Wallace. *The solution of quasi birth and death processes arising from multiple access computer systems.* PhD thesis, Systems Engineering Laboratory, University of Michigan, 1969.

	Matrix-sign Function Method	Schur Decomposition Method
1.	Define $N(d)$, $D(d)$, and f as in (23), (24), and (25).	
2.	Define the matrices A, E, and C as in (27), $\bar{N}$ as in (28), and $\bar{D}$ as in (29).	
3.	Define $W = (E + A)^{-1}(A - E)$, γ, μ as in (46), and W_e as in (45). Then find $S = sgn(W_e)$ by the quadratically convergent iteration (47) with the stopping criterion (48).	Find Θ and Ψ as in (58).
4.	• Find the rank-revealing QR decomposition of $S + I$ as in (50) and define V_1 and m_u as in (51). • Find the rank-revealing QR decomposition of $S - I$ as in (52) and define V_2 and m_s as in (53). • Find the rank-revealing QR decompositions in (54) and (55) and define U_1 and U_2 as in (56) and (57), respectively. • Define U and V as in (30).	Define $U = \Theta$ and $V = \Psi$ and partition U and V as in (30). Note that $U^{-1} = U^T$ here.
5.	Let E_{22} and A_{22} be the lower right $m_s \times m_s$ blocks of $U^{-1}EV$ and $U^{-1}AV$ as in (31), respectively. Then define F as in (36).	
6.	Define Z as in (39) and solve for x_0 and z_0 from (39) and (40).	
7.	Define L_2 as in (37) and also $g \triangleq z_0 U_2 \quad \text{and} \quad H \triangleq L_2 C$ to obtain the matrix-geometric expression $x_{k+1} = gF^k H,\ k \geq 0$.	

Table 1: Summary of the overall numerical algorithm for infinite M/G/1 type Markov chains.

		SSI Method					MSF Method			GSD Method	
f	m	T_{CPU}	Error	I_G	K_1	K_2	T_{CPU}	Error	I_S	T_{CPU}	Error
2	16	0.283	1.9e-09	37	21	20	0.317	8.8e-16	6	0.150	4.4e-15
	32	1.950	1.4e-09	37	20	20	2.133	1.8e-15	7	0.983	2.2e-14
	64	14.466	1.9e-09	36	20	20	19.049	6.2e-15	7	8.133	8.7e-14
3	16	0.350	1.5e-09	40	16	16	0.800	2.6e-14	6	0.367	7.5e-15
	32	2.217	1.7e-09	38	16	15	6.516	3.2e-14	6	2.600	1.2e-12
	64	16.516	1.8e-09	37	16	15	53.681	1.6e-13	6	22.732	6.2e-12
4	16	0.417	1.9e-09	42	15	14	1.817	4.7e-14	6	0.817	2.6e-13
	32	2.717	1.9e-09	39	15	14	15.366	6.2e-13	6	6.000	1.5e-12
	64	20.199	1.6e-09	38	14	14	124.612	3.9e-12	6	46.715	1.9e-11

Table 2: CPU time and error results for $\rho = 0.6$. I_G and I_S are the numbers of successive substitution iterations and matrix-sign function iterations, respectively. K_1 and K_2 are the truncation indices for the $\{A_k\}$ and $\{B_k\}$ (and also $\{\hat{A}_k\}$ and $\{\hat{B}_k\}$) matrix sequences, respectively.

		SSI Method					MSF Method			GSD Method	
f	m	T_{CPU}	Error	I_G	K_1	K_2	T_{CPU}	Error	I_S	T_{CPU}	Error
2	16	0.800	6.3e-10	165	27	26	0.333	6.0e-16	8	0.150	4.5e-15
	32	5.383	5.9e-10	162	27	26	2.283	1.0e-15	9	1.033	7.2e-15
	64	40.048	6.1e-10	160	27	26	19.833	2.9e-15	9	8.450	3.5e-14
3	16	1.083	5.9e-10	180	21	20	0.867	5.2e-15	8	0.367	1.9e-13
	32	7.333	6.1e-10	170	20	19	7.233	3.3e-14	8	2.617	5.2e-13
	64	53.498	5.7e-10	166	20	19	59.298	8.6e-14	8	22.266	2.8e-12
4	16	1.517	6.0e-10	194	18	18	1.933	6.8e-13	8	0.817	2.6e-13
	32	9.533	6.3e-10	177	18	17	16.749	2.3e-12	8	6.000	1.2e-12
	64	70.031	5.7e-10	170	18	17	137.111	8.0e-12	8	48.598	2.1e-10

Table 3: CPU time and error results for $\rho = 0.9$. I_G and I_S are the numbers of successive substitution iterations and matrix-sign function iterations, respectively. K_1 and K_2 are the truncation indices for the $\{A_k\}$ and $\{B_k\}$ (and also $\{\hat{A}_k\}$ and $\{\hat{B}_k\}$) matrix sequences, respectively.

	SSI Method					MSF Method			GSD Method	
ρ	T_{CPU}	Error	I_G	K_1	K_2	T_{CPU}	Error	I_S	T_{CPU}	Error
0.4	1.533	1.1e-09	22	13	13	6.533	1.7e-14	6	2.583	6.6e-13
0.6	2.267	1.7e-09	38	16	15	6.533	3.2e-14	6	2.600	1.2e-12
0.8	4.100	1.0e-09	84	19	18	6.866	2.5e-14	7	2.583	5.2e-13
0.9	7.233	6.1e-10	170	20	19	7.200	3.3e-14	8	2.650	5.2e-13
0.95	13.266	3.0e-10	335	21	20	7.550	1.2e-14	9	2.583	5.8e-13
0.99	56.514	5.9e-11	1542	21	21	8.516	1.6e-13	12	2.617	4.2e-14
0.995	107.096	2.9e-11	2946	21	21	8.483	7.9e-14	12	2.567	3.6e-13
0.999	-	-	-	-	-	9.433	1.7e-12	15	2.567	4.8e-13
0.9995	-	-	-	-	-	9.416	3.6e-12	15	2.583	3.0e-13
0.9999	-	-	-	-	-	11.100	3.7e-11	20	2.533	2.8e-12

Table 4: CPU time and error results as functions of ρ for $m = 32$ and $f = 3$. I_G and I_S are the numbers of successive substitution iterations and matrix-sign function iterations, respectively. K_1 and K_2 are the truncation indices for the $\{A_k\}$ and $\{B_k\}$ (and also $\{\hat{A}_k\}$ and $\{\hat{B}_k\}$) matrix sequences, respectively.

	$m = 16,\ f = 8$			$m = 32,\ f = 4$		
ρ	$K = 10$	$K = 100$	$K = 1000$	$K = 10$	$K = 100$	$K = 1000$
0.6	1.9e-15	2.5e-15	2.5e-15	2.5e-15	2.6e-15	2.6e-15
0.9	1.3e-14	6.6e-15	6.5e-15	4.7e-15	3.9e-15	4.0e-15
0.999	6.9e-11	9.6e-12	1.3e-12	2.7e-11	3.2e-12	5.9e-13
1.001	1.4e-10	2.0e-11	2.5e-12	5.2e-11	6.1e-12	1.0e-12
1.1	1.3e-14	5.0e-15	4.9e-15	1.8e-15	2.1e-15	2.1e-15
1.4	1.9e-15	1.7e-15	1.7e-15	1.1e-15	1.1e-15	1.1e-15
$T_{CPU}^{(1)}$	7.166	9.583	15.233	6.766	7.616	11.900
$T_{CPU}^{(2)}$	7.133	9.366	13.083	6.750	7.400	9.816
$T_{CPU}^{(3)}$	5.016			4.566		

Table 5: The error of the generalized Schur decomposition method for the finite M/G/1 chain as a function of ρ for $K = 10$, 100, and 1000. $T_{CPU}^{(1)}$ is the total time elapsed to obtain the complete solution vector. $T_{CPU}^{(2)}$ is the time elapsed to obtain the matrix-geometric factors for the finite chain, whereas $T_{CPU}^{(3)}$ is that for the corresponding infinite chain.

ρ	I_S	$P[empty]$	$E[queue]$	Error
0.6	10	7.525034e-1	3.321097e-1	3.8e-16
0.8	10	2.756323e-1	3.345204e0	2.3e-14
0.9	10	7.930838e-2	5.016590e3	3.4e-16
0.95	10	3.889021e-2	3.001338e4	1.7e-15
0.99	11	7.667718e-3	2.300500e5	1.2e-14
0.999	11	7.643891e-4	2.480478e6	9.0e-14

Table 6: The probability of empty queue, the expected queue length, and the error as functions of utilization. I_S is the number of matrix-sign function iterations. $\epsilon = 10^{-10}$ is used for the stopping criterion (48).