An Invariant Subspace Approach in M/G/1 and G/M/1 Type Markov Chains

Nail Akar and Khosrow Sohraby
Computer Science Telecommunications
University of Missouri - Kansas City
Kansas City, MO 64110

Abstract Let $A_k,\ k \geq 0$, be a sequence of $m \times m$ nonnegative matrices and let $A(z) = \sum_{k=0}^{\infty} A_k z^k$ be such that $A(1)$ is an irreducible stochastic matrix. The unique power-bounded solution of the nonlinear matrix equation $G = \sum_{k=0}^{\infty} A_k G^k$ has been shown to play a key role in the analysis of Markov chains of M/G/1 type. Assuming that the matrix $A(z)$ is rational, we show that the solution of this matrix equation reduces to finding an invariant subspace of a certain matrix. We present an iterative method for computing this subspace which is globally convergent. Moreover, the method can be implemented with *quadratic* or *higher* convergence rate *matrix sign function* iterations, which brings in a new dimension to the analysis of M/G/1 type Markov chains for which the existing algorithms may suffer from low linear convergence rates. The method can be viewed as a "bridge" between the matrix analytic methods and transform techniques whereas it avoids the requirement for a large number of iterations which may be encountered in the methods of the former type and the root finding problem of the techniques of the latter type. Similar results are obtained for computing the unique power-summable solution of the matrix equation $R = \sum_{k=0}^{\infty} R^k A_k$, which appears in the analysis of G/M/1 type Markov chains.

Keywords: M/G/1 and G/M/1 type Markov chains, nonlinear matrix equations, polynomial and rational matrices, matrix fractional descriptions, left and right invariant subspaces, matrix sign function.

1 INTRODUCTION

As shown by Neuts [41] and [42], a large class of probability models can be studied using the matrix generalizations of embedded Markov chains encountered in the basic M/G/1 and GI/M/1 queueing models. These are called the M/G/1 and G/M/1 type Markov chains, respectively. The development of efficient and reliable algorithms for the numerical solution of such models is extremely important to enlarge the class of probability models that can be solved by algorithmic methods [44]. Our interest in this paper is the study of irreducible discrete-time Markov chains of M/G/1 and G/M/1 type. The state-space of such a chain consists of integer pairs (i, j) where i is called the level of the chain and takes on an infinite set of values $i \geq 0$, while j is called the phase of the chain and takes on the finite set of values $0 \leq j \leq m-1$. The state transitions for a Markov chain of M/G/1 type, say $\mathcal{C}$, are assumed to satisfy the following properties. The transitions from level $i+1,\ i \geq 0$, to level $k+i$ are governed by the $m \times m$ matrix $A_k,\ k \geq 0$, whereas the transitions from the boundary level 0 to level k are given by the $m \times m$ matrix $B_k,\ k \geq 0$. Therefore, the transition matrix P of the Markov chain $\mathcal{C}$ takes the form

$$P = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & \cdots \\ A_0 & A_1 & A_2 & A_3 & \cdots \\ 0 & A_0 & A_1 & A_2 & \cdots \\ 0 & 0 & A_0 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{1}$$

We define

$$A(z) = \sum_{k=0}^{\infty} A_k z^k \tag{2}$$

and assume that $A(1)$ is an irreducible stochastic matrix. It is well-known that when $\mathcal{C}$ is ergodic, there exists a set of vectors $\pi = \begin{bmatrix} \pi_0 & \pi_1 & \cdots \end{bmatrix}$, where π_i is $1 \times m$, which satisfies the equation

$$\pi = \pi P,\ \pi e = 1, \tag{3}$$

in which $e = \begin{bmatrix} 1 & 1 & \cdots \end{bmatrix}^T$. Neuts [42] has given an algorithm to solve for (3) that is well-suited for numerical computation. Central to the matrix analytic approach of Neuts is the unique minimal nonnegative solution G_*

of the nonlinear matrix equation

$$G = A(G) = \sum_{k=0}^{\infty} A_k G^k. \tag{4}$$

It is also shown in [42] that, π_0 satisfies

$$\pi_0 = \pi_0 B(G_*) = \pi_0 \sum_{k=0}^{\infty} B_k G_*^k. \tag{5}$$

Once G_* is known, equation (5) can be used to calculate π_0. The vectors π_i's, $i \geq 1$, can then be recursively determined by a numerically stable algorithm in [42, pp.142-143].

The state space of a G/M/1 type Markov chain with its level and phase definitions is similar to that of the M/G/1 type. In this case, the transitions to the homogeneous level $i+1$, $i \geq 0$, from level $k+i$ are governed by the $m \times m$ matrices A_k's, $k \geq 0$, while the transition to the boundary level 0 from level k are given by the $m \times m$ matrices B_k's, $k \geq 0$. We then have the transition matrix $\bar{P}$ of a chain of G/M/1 type, say $\bar{\mathcal{C}}$, in the following form:

$$\bar{P} = \begin{bmatrix} B_0 & A_0 & 0 & 0 & \cdots \\ B_1 & A_1 & A_0 & 0 & \cdots \\ B_2 & A_2 & A_1 & A_0 & \cdots \\ B_3 & A_3 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{6}$$

The probability generating matrix $A(z)$ of this chain is defined in the same way as (2). The above chain has been also studied by Neuts [41] who developed an algorithmic method for its solution. The key role in his method is played by the matrix R_* which is the unique minimal nonnegative solution of the nonlinear matrix equation

$$R = \bar{A}(R) = \sum_{k=0}^{\infty} R^k A_k. \tag{7}$$

Neuts [41] has also shown that when $\bar{\mathcal{C}}$ is ergodic, the solution of

$$\pi = \pi \bar{P}, \ \pi e = 1, \tag{8}$$

has the matrix geometric form

$$\pi = \begin{bmatrix} \pi_0 & \pi_0 R_* & \pi_0 R_*^2 & \cdots \end{bmatrix} \tag{9}$$

where the $1 \times m$ vector π_0 is determined via the equations

$$\pi_0 = \pi_0 \bar{B}(R_*) = \pi_0 \sum_{k=0}^{\infty} R_*^k B_k, \tag{10}$$

$$\pi_0 (I - R_*)^{-1} e = 1, \tag{11}$$

where we use the notation I and e for the identity matrix and a column vector of ones of suitable sizes, respectively.

The transition matrices (1) and (6) are said to be in *canonical form* since all the matrix entries are square $m \times m$ [29]. Complex boundary behavior may also be introduced in the transition matrices that may change this canonical structure while still preserving the M/G/1 and G/M/1 structures. An extensive treatment of the cases with general boundaries is made in [41] and [42].

In what follows, we briefly review different approaches for the solution of the nonlinear matrix equations (4) and (7) arising in general Markov chains of M/G/1 and G/M/1 type. Basically, two approaches have been proposed to solve for (3) and (8): matrix-analytic methods and transform techniques. The former methods pioneered by Neuts [42],[41] mainly consist of the iterative algorithms

$$G_0 = 0,\ G_{j+1} = (I - A_1)^{-1} \left(\sum_{k \geq 0, k \neq 1}^{\infty} A_k G_j^k \right), \tag{12}$$

and

$$R_0 = 0,\ R_{j+1} = \left(\sum_{k \geq 0, k \neq 1}^{\infty} R_j^k A_k \right) (I - A_1)^{-1}, \tag{13}$$

to solve for $G = A(G)$ and $R = \bar{A}(R)$, respectively. Variants of the above-mentioned algorithms have been proposed in [44] and [25] to decrease the computational load (CPU time), however, most of these algorithms, despite the improvements achieved, fall into the class of algorithms with a linear convergence rate [44]. An exception to above is the logarithmic reduction algorithm of Latouche and Ramaswami [35] for finding the matrix-geometric rate matrix for QBD chains with quadratic convergence. With their results, it is now possible to analyze large scale QBD chains. Similar reduction techniques have also been used by Ye and Li [49],[50] for finite QBD chains in analysis of multi-media traffic queues and by Bini and Meini for general M/G/1 type Markov chains [7]. Quadratically convergent algorithms for the more general M/G/1 type Markov chains have recently been proposed in [34],[36], and [8], but an extensive numerical experimentation is still necessary to evaluate the performance of these new methods.

In matrix analytic methods, one needs the numerical evaluation and storage of matrices A_i's, $0 \leq i \leq K$, where K is the truncation index to make the numerical evaluation of the infinite summation at each iteration of the algorithms (12) and (13) feasible. When the matrices A_i converge slowly to the zero matrix as $i \to \infty$, choice of a large truncation index is required if numerical precision is sought. Besides the storage requirements, the use of a large truncation index definitely increases the number of matrix operations within each iteration of (12) and (13) and slows down the iterative process.

The other approach uses transform techniques to study Markov chains of M/G/1 and G/M/1 type, which in particular avoids the solution stage for the nonlinear matrix equations. With this approach, an expression for the generating function of the state probabilities is obtained which contains the unknown probability vector π_0. One then determines the zeros (roots) of the determinant

$$\Delta(z) = \det(zI - A(z)), \tag{14}$$

within the unit disk. These zeros are then used to obtain a set of linearly independent equations whose solution gives the desired unknown probabilities. Early work related with the transform techniques consists of [40] and [10] in the context of particular queueing problems. Recently, an extensive treatment of transform techniques is made in [17] and [19] for the general chains of M/G/1 and G/M/1 type. Neuts [41, pp.29-30] gives a set of reasons why transform techniques are algorithmically unattractive, which we summarize it as follows: i) it is difficult to compute the roots of $\Delta(z)$ accurately, ii) even though they can be computed, the roots are typically clustered together which makes the set of linear equations for solving π_0 ill-conditioned and an accurate computation of π_0 is then extremely difficult.

Our viewpoint is different from the two existing approaches. The basic assumption we make is that the probability generating matrix (pgm) $A(z)$ instead of being arbitrary, is a rational matrix in z. We note that most of the probability models lead to Markov chains with rational probability generating matrices. The truncation used in numerical evaluation of each iteration in the algorithms (12) and (13) can also implicitly be viewed as a method of approximating the pgm by a polynomial matrix which is indeed a special case of a rational matrix. In this paper, we show that in the case of a rational $A(z)$, significant reductions both in terms of CPU time and memory requirements can be achieved in solving the nonlinear matrix equations (4) and (7).

The key to our methodology in solving the nonlinear matrix equations arising in M/G/1 and G/M/1 type Markov chains is obtaining a polynomial matrix fractional description for the corresponding rational probability generating matrix $A(z)$. In other words, in the case of M/G/1 type chains, the rational matrix $A(z)$ is written as a fraction of two polynomial matrices, i.e., $A(z) = D^{-1}(z)N(z)$, for some *left coprime* polynomial matrices $D(z)$ and $N(z)$. For the G/M/1 type Markov chains, $A(z)$ is represented as a fraction of two *right coprime* polynomial matrices, i.e., $A(z) = P(z)Q^{-1}(z)$. Coprimeness condition (will be explained in Section 2) is imposed to avoid redundancies in the fractional description. The existence of such fractional descriptions for an arbitrary rational matrix $A(z)$ is well-known in *systems theory* and constructive procedures for obtaining these fractions are available in the literature, see for example Chen [12] and Kailath [28]. On the other hand, the problem descriptions for most of the related Markov chains in practice yield immediate fractional descriptions without any need for an additional constructive effort.

Once these fractions are obtained, the solution for the unique minimal nonnegative solution to the nonlinear matrix equations (4) and (7) (of possibly infinite terms) can be reduced to the unique power-bounded and power-summable solutions of the matrix *polynomial* equations

$$D(G)\,G = N(G) \quad \text{and} \quad R\,Q(R) = P(R), \tag{15}$$

respectively. In other words, via this reduction procedure, the truncation requirement in the numerical implementation process is easily avoided using the polynomial matrix fractional representation for the corresponding pgm. Therefore, in the case of a rational $A(z)$, one can achieve significant reductions in storage requirements and moreover avoid possible losses in accuracy due to truncation.

We introduce the use of *invariant subspaces* of real matrices to solve matrix polynomial equations of the form (15). For M/G/1 type Markov chains, we show that obtaining the unique power-bounded solution for

$D(G)\,G = N(G)$ amounts to finding an invariant subspace of an $mf \times mf$ real matrix in block companion form where $f = \max(\deg(D)+1, \deg(N))$. Here, deg denotes the *degree* operator on polynomial matrices [12] and m is the number of phases of the Markov chain. Moreover, the subspace of interest can easily be computed with the *matrix sign function* iterations, the theory of which is well-established in the context of solving the Lyapunov and the algebraic Riccati equation of *control theory* [37]. The matrix sign function can be computed iteratively by *quadratic* or *higher* convergent rate algorithms which brings in a new dimension to the analysis of M/G/1 type Markov chains for which the existing algorithms may suffer from low linear convergence rates. Similar results are also obtained for Markov chains of G/M/1 type. We note that the matrices of interest at each matrix sign function iteration are of size $mf \times mf$ in contrast to $m \times m$ matrices of the algorithms (12) and (13) but with the advantage of fewer number of iterations required, fewer number of matrix operations required at each step of the procedure, and suitability for *parallel* implementation [43].

Numerical stability is a critical issue in every proposed numerical algorithm. However, a theoretical analysis of stability of algorithms we propose for M/G/1 and G/M/1 type Markov chains is beyond the scope of the current paper. We refer to [3] and [4] on the numerical stability of matrix sign iterations based on which the algorithms are developed.

The organization of the paper is as follows. In Section 2, mathematical preliminaries necessary for the development of the paper are presented, including polynomial matrices, polynomial matrix fractional descriptions of rational matrices, and invariant subspaces of real matrices. Section 3 is devoted to the matrix sign function and its computation which will play a key role in our algorithms. In Section 4, we give examples from queueing theory on how to obtain the left and right coprime polynomial matrix fractional representations of $A(z)$. The nonlinear matrix equations (4) and (7) and their reductions to matrix polynomial equations of the form (15) are examined in Sections 5 and 6, respectively. In Section 7, we present our numerical algorithms for M/G/1 and G/M/1 type Markov chains based on the matrix sign function iterations. Section 8 covers a set of numerical examples to demonstrate the effectiveness of our approach. The final section is devoted to conclusions.

2 PRELIMINARIES AND NOTATION

In this section, we first focus on rational matrices of the form

$$A(z) = \{a_{i,j}(z)\},$$

where $a_{i,j}(z)$ is a rational function of the indeterminate z, i.e.,

$$a_{i,j}(z) = \frac{p_{i,j}(z)}{q_{i,j}(z)},$$

for some polynomial pairs $p_{i,j}(z)$ and $q_{i,j}(z)$. The following results on polynomial matrix fractions of the rational matrix $A(z)$ are based on [12].

A polynomial matrix is a matrix with polynomial entries. Let $Q(z)$ and $P(z)$ be $q \times q$ and $p \times q$ polynomial matrices, and let $Q(z)$ be nonsingular in the field of rational functions. Then $Q(z)$ and $P(z)$ are called *right coprime* if and only if for every z in $\mathbf{C}$ (the field of complex numbers), the matrix

$$\begin{bmatrix} Q(z) \\ P(z) \end{bmatrix}$$

has rank q. Similarly, two polynomial matrices $D(z)$ and $N(z)$ of size $p \times p$ and $p \times q$, respectively, are called *left coprime* if and only if for every z in $\mathbf{C}$, the matrix

$$\begin{bmatrix} D(z) & N(z) \end{bmatrix}$$

has rank p. Consider a rational matrix $A(z)$ of size $p \times q$. The fraction $A(z) = P(z)Q^{-1}(z)$ is called a right coprime polynomial fraction if the polynomial matrices $Q(z)$ and $P(z)$ are right coprime and $A(z) = D^{-1}(z)N(z)$ is a left coprime polynomial fraction if $D(z)$ and $N(z)$ are left coprime. Either fraction is also called an irreducible fraction. Given a rational matrix $A(z)$, it is always possible to obtain irreducible fractions [12].

We now summarize some notations and results concerning the invariant subspaces. More details on what follows can be found in the references [32] and [23]. Let us begin with some notation on linear spaces, subspaces, and matrices. By $\mathbf{R}^m$ we mean the real linear space of column vectors of m real numbers. Then $\mathbf{R}^{m \times n}$ is the linear space of $m \times n$ matrices with real entries. A *subspace* is a subset of $\mathbf{R}^m$ that is also closed under the operations of addition and scalar multiplication. For arbitrary subspaces

$\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathcal{S}_1 \subset \mathcal{S}_2$ denotes either inclusion or equality. If $A \in \mathbf{R}^{m\times n}$, we define the image of A by

$$\text{Im } A = \{x \in \mathbf{R}^m | x = Ay \text{ for some } y \in \mathbf{R}^n\}.$$

If Im A = Im B, then there exists a nonsingular matrix U such that $B = AU$.

The set of all eigenvalues of a matrix $A \in \mathbf{R}^{m\times m}$ is called the *spectrum* of A and written $\sigma(A)$. Let C be a region in the complex plane. Then the matrix A is called C-*stable* if $\sigma(A) \subset C$. Let C^l, C^0, and C^r denote the left-half plane, the imaginary axis, and the right half-plane of the complex plane, respectively. We also use the notation C^{cl} to denote closed left-half plane, i.e., $C^{cl} = C^l \cup C^0$. A subspace $\mathcal{S}$ of $\mathbf{R}^m$ is said to be A *invariant* where $A \in \mathbf{R}^{m\times m}$, if $A\mathcal{S} \subset \mathcal{S}$. Here, $A\mathcal{S} = \{x \in \mathbf{R}^m \ | x = Ay \text{ for some } y \in \mathcal{S}\}$. If a k-dimensional subspace $\mathcal{S}$ is A invariant, then $AS = SA_1$ holds for a $k \times k$ matrix A_1 and an $m \times k$ matrix S whose columns form a basis for $\mathcal{S}$, i.e., $\mathcal{S}$ = Im S. We call the A invariant subspace $\mathcal{S}$ the C-invariant subspace of A if $\sigma(A_1) \subset C$ and there is no larger A invariant subspace for which this inclusion holds. The dimension of $\mathcal{S}$ is the number of eigenvalues of A lying in C, counting multiplicities. We call the C^l-invariant subspace (C^r-invariant subspace) as *left* (*right*) invariant subspace for convenience of notation.

Based on [18], we have the following definitions. We say the matrix A is *power-bounded* if the set of powers $A^k,\ k \geq 0$, is bounded, i.e., sup $||A^k|| < \infty$, where $|| \cdot ||$ is any suitable matrix norm [24]. The matrix A is *power-summable* if the set of powers $A^k,\ k \geq 0$, is summable, i.e., $\sum_k ||A^k|| < \infty$. The spectrum of a power-bounded (power-summable) matrix lies in the closed unit disk (open unit disk). We use the notation A^T for the transpose of the matrix A and A^{-T} for $(A^{-1})^T$. We use I and e to denote the identity matrix and the column vector of ones, respectively, of appropriate sizes. A matrix with the property $A^k = 0$ for some integer $k \geq 0$ is called a nilpotent matrix.

The analysis of matrix algorithms and stopping criteria of algorithms require the use of matrix norms. In this paper, we use the 1- and ∞-norms of matrices. For a matrix $A \in \mathbf{R}^{m\times n}$, these norms are defined as [24]:

$$|| A ||_1 = \max_{1\leq j\leq n} \sum_{i=1}^{m} | a_{ij} | \quad \text{and} \quad || A ||_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{n} | a_{ij} |.$$

We show in Sections 5 and 6 that, solving the nonlinear matrix equations (4) and (7) arising in Markov chains of M/G/1 and G/M/1 type reduces to computing a basis for the left invariant subspace of a certain real matrix. In the following section, we will present one effective approach, namely the matrix sign function approach for finding the left invariant subspace of a real matrix which will play a dominant role in our algorithms presented in next sections.

3 LEFT INVARIANT SUBSPACE COMPUTATION VIA MATRIX SIGN ITERATIONS

In the past, the matrix sign function has been used to solve some important equations in control theory, such as Riccati or Lyapunov equations [15],[45],[9],[20] employing left invariant subspace computations. In this paper, we will use the same concept in another field, namely applied probability, in the context of numerical solutions of the nonlinear matrix equations of the form (4) and (7).

We first provide some introductory material regarding the definition and simple properties of the matrix sign function based on the references [45] and [37].

Definition 1 *Let* $M \in \mathbf{R}^{m\times m}$ *with no pure imaginary eigenvalues, i.e.,* $\sigma(M) \cap C^0 = \emptyset$. *Let* M *have a Jordan decomposition* $M = T(D+N)T^{-1}$ *where* $D = \mathrm{diag}\,\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ *and* N *is nilpotent and commutes with* D. *Then the matrix sign of* M *is given by*

$$Z = sgn(M) := T\ \mathrm{diag}\,\{sgn(\lambda_1), sgn(\lambda_2), \ldots, sgn(\lambda_m)\}\ T^{-1}, \qquad (16)$$

where for a complex scalar z *with* $\mathrm{Re}\ z \neq 0$, *the sign of* z *is defined by*

$$sgn(z) = \begin{cases} 1 & \textit{if Re } z > 0 \\ -1 & \textit{if Re } z < 0 \end{cases}$$

Here are some elementary and easily obtainable properties of $Z = sgn(M)$ [37]:

1. Z is diagonalizable with eigenvalues ± 1, $Z^2 = I$, and $Z = Z^{-1}$,
2. Im $(Z - I) =$ left invariant subspace of M,

3. Im $(Z + I)$ = right invariant subspace of M,

4. For all scalars c, $sgn(cM) = sgn(c)sgn(M)$.

5. $sgn(TMT^{-1}) = Tsgn(M)T^{-1}$ for any nonsingular T.

By property 2, an orthogonal basis for the left invariant subspace of M which has a dimension r is given by the first r columns of the orthogonal matrix in a rank-revealing QR decomposition of $(Z - I)$ [11]. Alternative methods are presented in [3] and [4].

The above definition for the matrix sign does not lend itself to an efficient computation but there are several ways of evaluating the matrix sign function. The simplest iteration scheme is Newton's method applied to $sgn(M)^2 = I$:

$$Z_0 = M,\ Z_{k+1} = \frac{1}{2}(Z_k + Z_k^{-1}). \tag{17}$$

Then

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

This iteration is called the classical matrix sign function algorithm and the iteration converges quadratically for all matrices M for which the matrix sign is well-defined [45]. There are also ways of accelerating iterations with scaling. Two popular ways of scaling Newton's scheme are

$$Z_{k+1} = \alpha_k Z_k + (1 - \alpha_k)Z_k^{-1}, \tag{18}$$

and

$$Z_{k+1} = \frac{1}{2}(\gamma_k Z_k + \gamma_k^{-1} Z_k^{-1}), \tag{19}$$

where α_k and γ_k are appropriately chosen scalars. Two simple scaling schemes based on the determinant of the intermediate matrices which have proven to be useful are

$$\begin{aligned} \text{Balzer [6]}: \ \alpha_k &= (|\det Z_k|^{1/m} + 1)^{-1}, && (20)\\ \text{Byers [9]}: \ \gamma_k &= |\det Z_k|^{-1/m}. && (21) \end{aligned}$$

A comparison of different scaling schemes for different α_k is presented by Balzer [6], and for different γ_k by Kenney and Laub [31]. In the implementations of the scaling schemes (20) and (21), we note that the determinant $\det Z_k$ can be calculated from the LU factors that we use to perform the matrix inversion at each step [3] so that this kind of scaling does not introduce additional complexity to the classical scheme (17). The stopping criterion we will employ is the one used by [3]:

$$\| Z_{k+1} - Z_k \|_1 < \varepsilon \| Z_k \|_1, \tag{22}$$

where ε is a small, user specified error bound.

More general iteration schemes for computing the matrix sign are introduced by Kenney and Laub [30] in which a rational approximation characterized by a parameter pair (n, l) is used for obtaining the following matrix iteration

$$Z_{k+1} = Z_k \, p_{n,l}(I - Z_k^2) \, [q_{n,l}(I - Z_k^2)]^{-1}. \tag{23}$$

In the above iteration, $p_{n,l}$ and $q_{n,l}$ are polynomials of order n and l, respectively. Defining $(\alpha)_k = \alpha(\alpha+1)\cdots(\alpha+k-1)$ with $(\alpha)_0 = 1$, these polynomials are given as [30]

$$p_{n,l}(x) = \sum_{k=0}^{n} \frac{(\frac{1}{2})_k \, (\frac{1}{2} - l)_l \, (k - n - l)_l}{k! \, (-n - l)_l \, (k + \frac{1}{2} - l)_l} x^k,$$

and

$$q_{n,l}(x) = \sum_{k=0}^{l} \frac{(-l)_k \, (-\frac{1}{2} - n)_k}{k! \, (-n - l)_k} x^k.$$

The appropriate matrix-valued iteration formulas can then be directly written by Table 1 taken from [30] for $0 \le n, l \le 2$. The table should be

l/n	0	1	2
0	x	$\frac{x(3-x^2)}{2}$	$\frac{x(15-10x^2+3x^4)}{8}$
1	$\frac{2x}{1+x^2}$	$\frac{x(3+x^2)}{1+3x^2}$	$\frac{x(15+10x^2-x^4)}{4(1+x^2)}$
2	$\frac{8x}{3+6x^2-x^4}$	$\frac{4x(1+x^2)}{1+6x^2+x^4}$	$\frac{x(5+10x^2+x^4)}{1+10x^2+5x^4}$

Table 1: Direct rational iteration formulas for the matrix sign function.

read as follows. Suppose we use the approximation corresponding to the case $n = 0$ and $l = 1$, then the associated entry in the table $2x/(1 + x^2)$ yields the following direct iteration

$$Z_{k+1} = 2 \, Z_k \, (I + Z_k^2)^{-1}. \tag{24}$$

This particular iteration is called the inverse classical matrix sign function algorithm since the matrices obtained at each iteration are the inverses of those obtained by the classical iteration (17). It has also been shown in [30] that, the iterations (23) are globally convergent when $n = l$ (excluding the case $n = l = 0$) or $n = l - 1$ and the convergence rate is of order $(n + l + 1)$. For other cases, global convergence may not be guaranteed which in particular suggests that inverse-free iterations corresponding to the case $l = 0$ do not have global convergence. Comparing the general rational iterations for the matrix sign function with the Newton iteration with scaling, the tradeoff is the larger number of flops per iteration versus the higher convergence rates. On a final remark on the rational iterations for the matrix sign function, these iterations are suitable for parallel implementation. Pandey, Kenney, and Laub [43] present an explicit partial fraction form for the iterations when $n = l - 1$. The iteration is

$$Z_{k+1} = \frac{1}{l}\, Z_k \sum_{i=1}^{l} \frac{1}{\beta_i}\, (Z_k^2 + \alpha_i^2 I)^{-1}, \tag{25}$$

where

$$\beta_i = \frac{1}{2}\left(1 + \cos\frac{(2i-1)\pi}{2l}\right),\ \alpha_i^2 = \frac{1}{\beta_i} - 1,\ 1 \le i \le l.$$

As shown in [43], this iteration is suitable for parallel implementation since each of the l terms in the expansion (25) can be evaluated on a separate processor in parallel. It has been shown in [43] that, i Newton steps of the form (17) give the inverse of the of the $(l-1, l)$ approximation (25) where $l = 2^{i-1}$. Therefore, one may expect significant improvements in computation times in using $(l - 1, l)$ approximations and their parallel implementation. For parallel implementations of different matrix sign iterations, see [21],[26], [27],[5].

We will use Newton's iteration with the scaling scheme (21) in our numerical examples since it is widely accepted [37] together with the rank revealing QR decomposition to extract the left invariant subspace form the matrix sign. A detailed study of performance assessment of different matrix sign function iterations is beyond the scope of this paper. For a brief survey on different techniques for computing left or right invariant subspaces including the Schur decomposition based methods, we refer the reader to [3] and [4].

4 THE CASE OF RATIONAL $A(z)$

We assume that the probability generating matrix $A(z)$ is a rational matrix, and based on Section 2, we have the existence of polynomial matrices Q, P, D, and N such that $A(z)$ is represented by the irreducible polynomial fractions

$$A(z) = P(z)Q^{-1}(z) = D^{-1}(z)N(z). \tag{26}$$

Many of the Markov chains encountered in applied probability yield rational probability generating matrices. Below, we present several examples from the literature.

1. Quasi-birth-and-death-processes [48],[41],[33]

If in the structure of P in (1) or $\bar{P}$ in (6), changes of states are restricted to adjacent levels, those Markov chains are of both M/G/1 and G/M/1 type and named as quasi-birth-and-death-processes. In this case

$$A_k = 0, \; k > 2,$$

and

$$A(z) = A_0 + A_1 z + A_2 z^2,$$

and one can immediately obtain the fractions for $A(z)$ through the natural choice of

$$Q(z) = D(z) = I, \;\; P(z) = N(z) = A(z).$$

Besides, if $A_k = 0, \; k > K, \; 2 < K < \infty$, then $A(z)$ is still a polynomial matrix and the choices above for the irreducible fractions remain the same for this case. In [42], K is called the truncation index which is chosen appropriately so that the assumption $A_k \approx 0, \; k > K$ does not result in an intolerable inaccuracy.

2. Single server discrete-time queue with modulated arrivals

Consider a discrete-time queue with a single server with arrivals modulated by a finite state discrete-time Markov chain [13],[47]. The modulating Markov chain has m states with transition probabilities $q_{i,j}, \; 0 \leq i,j \leq m-1$. Transitions are assumed to occur at slot boundaries. The probability of k arrivals when the underlying Markov chain resides in state i is denoted $h_{i,k}$. Assume that

$$h_i(z) = \sum_{k=0}^{\infty} h_{i,k} z^k \tag{27}$$

is a rational function of z (for example, a discrete phase-type distribution). Let the queue length and state of the underlying Markov chain be associated with our level and phase definition. Then if we write

$$h_i(z) = \frac{n_i(z)}{q_i(z)},\ 0 \le i \le m-1,$$

for coprime polynomials $n_i(z)$ and $q_i(z)$, the pgm $A(z)$ can be written as

$$A(z) = Q_0 \operatorname{diag}\{h_i(z)\} = Q_0\, N(z) Q^{-1}(z) = Q_0\, Q^{-1}(z) N(z)$$

where $Q_0 = \{q_{i,j}\}, N(z) = \operatorname{diag}\{n_i(z)\}$, and $Q(z) = \operatorname{diag}\{q_i(z)\}$. Moreover, if we define

$$P(z) = Q_0\, N(z),\ D(z) = Q(z) Q_0^{-1},$$

we obtain the irreducible fractions (26) for $A(z)$.

3. MMPP/G/1 queue [38]:

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 $R = \operatorname{diag}\{\lambda_i\},\ 0 \le i \le m-1$. 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

$$h(s) = \frac{p(s)}{q(s)}$$

for some coprime polynomials $p(s)$ and $q(s)$. If we consider the embedded Markov renewal process at departure epochs, we obtain a Markov chain of M/G/1 type where

$$A(z) = h(\Lambda - R - \Lambda z),$$

which is a rational function of z [39]. The polynomial fractions of $A(z)$ can directly be obtained by defining

$$Q(z) = D(z) = q(\Lambda - R - \Lambda z),\quad P(z) = N(z) = p(\Lambda - R - \Lambda z).$$

The above examples demonstrate that for many queueing problems of practical interest, polynomial fractional representations of $A(z)$ can easily be obtained without the need for evaluating the matrices A_i's, $i \ge 0$. Hereafter we assume that $A(z)$ is rational and has the left and right fractions given in (26). In the next two sections, we will study the nonlinear equations (4) and (7).

5 THE EQUATION $G = A(G)$

Before studying the nonlinear matrix equation $G = A(G)$, we need the following technical assumptions:

(A1) The matrix $A(z)$ is analytic in the closed unit disk,

(A2) The determinant $\Delta(z)$ defined in (14) does not vanish on the unit circle except at $z = 1$.

A key parameter that determines the number of zeros of $\Delta(z)$ in the unit disk is

$$\gamma = \frac{d}{dz}\Delta(z)|_{z=1}.$$

For the sake of completeness in our analysis, we restate the following result given in [17] on the number of such zeros.

Theorem 1 *Let assumptions (A1) and (A2) hold.*

(i) If $\gamma > 0$, then $\Delta(z)$ has exactly $m-1$ zeros in the open unit disk and a simple zero at $z = 1$.

(ii) If $\gamma < 0$, then $\Delta(z)$ has exactly m zeros in the open unit disk.

In [19], the authors show for a Markov chain of M/G/1 type (with the abovementioned assumptions on $A(z)$) that, when $\gamma > 0$, the chain is ergodic and moreover, there is a unique power-bounded solution to $G = A(G)$ and G_* is the unique power-bounded solution. Similarly, in [19] it is shown that, a Markov chain of G/M/1 type is ergodic if $\gamma < 0$ and when the Markov chain is ergodic, there is a unique power-summable solution to $R = \bar{A}(R)$ and R_* is the unique power-summable solution. Hereafter we assume an ergodic M/G/1 type Markov chain in this section.

We now recall the existence of a left-coprime fraction of $A(z)$:

$$A(z) = D^{-1}(z)N(z), \tag{28}$$

where

$$D(z) = D_0 + D_1 z + \cdots + D_d z^d, \quad N(z) = N_0 + N_1 z + \cdots + N_n z^n,$$

are assumed to be polynomial matrices of degrees d and n, respectively. We also have the assumptions (A1), (A2), and $\gamma > 0$ so that the Markov chain is ergodic and $\Delta(z)$ has exactly $m-1$ zeros in the unit circle and one simple zero at $z=1$. Recall that with these assumptions $G = A(G)$ has a unique power-bounded solution G_*. Let us now define

$$F(z) = zD(z) - N(z) = \sum_{i=0}^{f} F_i z^i, \tag{29}$$

where

$$f = \max(d+1, n). \tag{30}$$

We note that the degree f of the polynomial matrix $F(z)$ is central in evaluating the computational complexity of the algorithms we propose. Note by (28) and (29) that

$$F(z) = D(z)[zI - A(z)]. \tag{31}$$

Since $A(z)$ is analytic in the closed unit disk and the polynomial matrices $D(z)$ and $N(z)$ are left coprime, it is immediate to see that [12] $\det D(z)$ has all its roots outside the unit disk by (A1). It then follows by (31) that the roots of $\Delta(z)$ and $\det F(z)$ within the unit disk coincide, therefore $\det F(z)$ has $m-1$ roots in the unit disk and one simple zero at $z=1$. We also define

$$F(G) = \sum_{i=0}^{f} F_i G^i$$

so as to state the following result.

Theorem 2 *G_* satisfies the matrix polynomial equation*

$$F(G) = 0. \tag{32}$$

Proof. We have $G_* = A(G_*)$. Premultiply the equality by $D(G_*)$ so that $D(G_*)G_* = D(G_*)A(G_*)$. Then it remains to show that $D(G_*)A(G_*) = N(G_*)$. To see that

$$\begin{aligned} D(G_*)A(G_*) &= D_0 A(G_*) + D_1 G_* A(G_*) + \cdots + D_d G_*^d A(G_*), \\ &= D_0 A(G_*) + D_1 A(G_*) G_* + \cdots + D_d A(G_*) G_*^d, \\ &= \sum_{i=0}^{\infty} \sum_{j=0}^{i} D_j A_{i-j} G_*^i, \end{aligned} \tag{33}$$

where $D_j = 0,\ j > d$ and the second equality comes from $G_* = A(G_*)$. Considering the fraction (28) we also have

$$D(z)A(z) = N(z),$$

which immediately yields

$$N(z) = \sum_{i=0}^{\infty} \sum_{j=0}^{i} D_j A_{i-j} z^i, \tag{34}$$

for which $N_j = 0,\ j > n$. Comparing (33) and (34) we obtain

$$D(G_*)A(G_*) = N(G_*)$$

or $D(G_*)G_* = N(G_*)$. By definition of $F(G)$, it is now clear that $F(G_*) = 0$. •

We now define a new polynomial matrix with the indeterminate s in the following way:

$$H(s) = (1-s)^f F(z)|_{z=\frac{1+s}{1-s}}. \tag{35}$$

Actually,

$$H(s) = \sum_{i=0}^{f} H_i s^i = \sum_{i=0}^{f} F_i (1-s)^{f-i}(1+s)^i \tag{36}$$

has the following properties:

- H_f is nonsingular. On a closer look at (35), one can write

$$\begin{aligned} H_f &= \lim_{s\to\infty} s^{-f} H(s), \\ &= \lim_{s\to\infty} s^{-f}(1-s)^f F(-1), \\ &= (-1)^f F(-1). \end{aligned}$$

  Since $\det F(z)$ has only a simple zero at $z = 1$ on the unit disk, $F(-1)$ is nonsingular, so is H_f.

- Since H_f is nonsingular, $\det H(s)$ has mf roots. From (35), it is easy to see that s_0 is a root of $\det H(s)$ if and only if $\det F(z_0) = 0$ and $s_0 = \frac{z_0-1}{z_0+1}$ or $s_0 = 1$. Since the map $\frac{z-1}{z+1}$ transforms the open unit disk to C^l, and the unit circle to C^0, we conclude that $\det H(s)$ has exactly $m-1$ roots in C^l and a simple root at $s = 0$.

We also recall that the eigenvalues of G_* are exactly the roots of $\Delta(z)$ in the closed unit disk [19], therefore in particular, G_* cannot have an eigenvalue at $\lambda = -1$ and the matrix inverse $(I + G_*)^{-1}$ exists. Let us then define

$$G_0 = (G_* - I)(G_* + I)^{-1}. \tag{37}$$

It is easy to see that G_0 is a C^{cl}-stable matrix, having one simple eigenvalue $\lambda = 0$ on the imaginary axis . Defining

$$H(G) = \sum_{i=0}^{f} H_i G^i$$

we have the following result.

Theorem 3 *G_0 is the unique C^{cl}-stable solution for the matrix equation*

$$H(G) = 0.$$

Proof. First, we show $H(G_0) = 0$. Since G_0 is C^{cl}-stable, it cannot have an eigenvalue at $\lambda = 1$, therefore $(I - G_0)$ is nonsingular and moreover G_* is related to G_0 via

$$G_* = (I + G_0)(I - G_0)^{-1}. \tag{38}$$

Then by using (38), one can write

$$F(G_*) = \sum_{i=0}^{f} F_i (I + G_0)^i (I - G_0)^{-i}. \tag{39}$$

Postmultiply (39) by $(I - G_0)^f$ so that we obtain

$$\begin{aligned} F(G_*)(I - G_0)^f &= \sum_{i=0}^{f} F_i (I + G_0)^i (I - G_0)^{f-i}, \\ &= \sum_{i=0}^{f} H_i G_0^i. \text{ by (36)} \end{aligned}$$

But since $F(G_*) = 0$, so is $H(G_0)$. $\bullet$

The uniqueness part deserves a particular attention since the development here gives a constructive procedure for evaluating G_*. For this purpose we first define

$$\hat{H}_i = H_f^{-1} H_i, \; 0 \le i \le f - 1. \tag{40}$$

Let us then write G_0 in the form VJV^{-1} where J is in Jordan canonical form and $V = \begin{bmatrix} V_0 & V_1 & \cdots & V_{m-1} \end{bmatrix}$ is a Jordan canonical basis for G_0. Define the matrices

$$E = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & I \\ -\hat{H}_0 & -\hat{H}_1 & -\hat{H}_2 & \cdots & -\hat{H}_{f-1} \end{bmatrix}, \quad V_e = \begin{bmatrix} V \\ VJ \\ VJ^2 \\ \vdots \\ VJ^{f-1} \end{bmatrix}. \tag{41}$$

By direct substitution one can show that

$$EV_e = V_e J.$$

Since V is nonsingular, the columns of V_e are linearly independent. It is also shown in [22] that the eigenvalues of E are exactly the singularities of $H(s)$, so E has m eigenvalues in C^{cl}. We therefore conclude that Im V_e is the C^{cl}-invariant subspace of E.

Let us assume the existence of another C^{cl}-stable matrix G_1 satisfying $H(G) = 0$. Let us also write G_1 in the form $V_1 J_1 V_1^{-1}$ where J_1 is in Jordan canonical form and $V_1 = \begin{bmatrix} V_{1,0} & V_{1,1} & \cdots & V_{1,m-1} \end{bmatrix}$ is a Jordan canonical basis for G_1. Defining

$$V_{1e} = \begin{bmatrix} V_1 \\ V_1 J_1 \\ V_1 J_1^2 \\ \vdots \\ V_1 J_1^{f-1} \end{bmatrix},$$

it holds that

$$EV_{1e} = V_{1e} J_1.$$

We thus conclude that Im V_{1e} is the C^{cl}-invariant subspace of E. Thus, by Section 2, there exists a nonsingular transformation U such that $V_{1e} = V_e U$ and $J_1 = U^{-1} J U$. This, in particular yields $V_1 = VU$. We then immediately write

$$G_1 = V_1 J_1 V_1^{-1} = VUU^{-1}JUU^{-1}V^{-1} = G_0.$$

which concludes the proof showing uniqueness of the C^{cl}-stable solution for $H(G) = 0$. •

We now give a constructive procedure to compute G_*, the unique power-bounded solution to $G = A(G)$ using theorems 1, 2, and 3. For this purpose, we find the C^{cl}-invariant subspace of the matrix E, say $\mathrm{Im}\, T$, which satisfies

$$ET = TE_1,$$

where $E_1 \in \mathbf{R}^{m\times m}$ and $\sigma(E_1) \subset C^{cl}$. Noting that T is $mf \times m$, we partition T into $m \times m$ blocks

$$T = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_f \end{bmatrix}. \tag{42}$$

The following theorem gives a new characterization of G_* in terms of the C^{cl}-invariant subspace of E and is essential to a numerical computation of G_*.

Theorem 4 *The matrix T_1 is nonsingular and the matrix X defined by $X = T_2T_1^{-1}$ equals G_0. Furthermore, the matrix $T_1 - T_2$ is nonsingular and the matrix Y defined by $Y = (T_1+T_2)(T_1-T_2)^{-1}$ is the unique power-bounded solution to $G = A(G)$, i.e., $Y = G_*$.*

Proof. Since $\mathrm{Im}\, T = \mathrm{Im}\, V_e$, there exists a nonsingular matrix U that $T = V_eU$ which yields $T_1 = VU$ showing the nonsingularity of T_1. By direct substitution, it is easy to show that $X = T_2T_1^{-1}$ satisfies $H(X) = 0$ which by uniqueness of the C^{cl}-stable solution, yields $X = G_0$. Since $T_1 - T_2 = (I - X)T_1$ and $I - X$ is nonsingular (since X is C^{cl}-stable and therefore does not have an eigenvalue at $\lambda = 1$), so is $T_1 - T_2$. Defining $Y = (T_1+T_2)(T_1-T_2)^{-1}$, it is straightforward to see that $Y = (I+G_0)(I-G_0)^{-1}$ which equals G_* by equation (38). •

We now have shown that the evaluation of G_* reduces to computing the C^{cl}-invariant subspace of the matrix E. At a first glance, one might object on the use of this equivalence for computation of G_*. This is because if the set C in the complex plane is arbitrary then the computation of the C-invariant subspace of E amounts to finding the generalized eigenvectors of the matrix E associated with its eigenvalues lying in C. However, the set C^{cl} is not an arbitrary set and deserves a particular attention. In Section 7, we will present iterative methods to compute the C^{cl}-invariant subspace of the matrix E without any need for calculating the individual eigenvalues of E.

6 THE EQUATION $R = \bar{A}(R)$

In this section, we study the unique power-summable solution R_* to the nonlinear matrix equation $R = \bar{A}(R)$, when assumptions (A1), (A2) hold and $\gamma < 0$ so that the associated Markov G/M/1 type Markov chain is ergodic. By theorem (1), this yields that $\Delta(z)$ has exactly m zeros in the unit disk. In [18], it is shown that the eigenvalues of R_* coincide with these m zeros. Observe that, transposing $R = \bar{A}(R)$ yields an equation similar in form to $G = A(G)$, namely

$$R^T = \sum_{k=0}^{\infty} A_k^T (R^T)^k,$$

so the development in Section 5 can easily be extended to the solution of $R = \bar{A}(R)$. The results for this case are presented below, but we will not give detailed proofs of these, since the methods used to obtain them are exactly the same as used for the equation $G = A(G)$. Let us then start with a right-coprime polynomial matrix fraction for $A(z)$:

$$A(z) = P(z)Q^{-1}(z), \tag{43}$$

where

$$P(z) = P_0 + P_1 z + \cdots + P_p z^p, \quad Q(z) = Q_0 + Q_1 z + \cdots + Q_q z^q,$$

are assumed to be polynomial matrices of degrees p and q, respectively. We then define

$$\bar{F}(z) = zQ(z) - P(z) = \sum_{i=0}^{\bar{f}} \bar{F}_i z^i, \; \bar{f} = \max(q+1, p) \tag{44}$$

so as to state the following theorem, an analogue of theorem 2.

Theorem 5 *R_* satisfies the matrix equation*

$$\bar{F}(R) = \sum_{i=0}^{\bar{f}} R^i \bar{F}_i = 0.$$

We then define the polynomial matrix $\bar{H}(s)$ in the same way as (35):

$$\bar{H}(s) = (1-s)^{\bar{f}} \bar{F}(z)|_{z=\frac{1+s}{1-s}}. \tag{45}$$

We also define

$$R_0 = (R_* - I)(I + R_*)^{-1}, \tag{46}$$

so that we have the following result, an analogue of theorem 3.

Theorem 6 *R_0 is the unique C^l-stable solution of the matrix equation*

$$\bar{H}(R) = \sum_{i=0}^{\bar{f}} R^i \bar{H}_i = 0.$$

Note. The only difference from the proof of theorem 3 is the fact that R_0 is C^l-stable and therefore it does not have any eigenvalue on the imaginary axis. This is because the transformation $(z-1)/(z+1)$ transforms the open unit disk to C^l, the open-left half plane, and that the matrix R_* is power-summable and has all its eigenvalues in the open unit disk.

Before giving the main result of this section, we need the following definitions.

$$\tilde{H}_i = [\bar{H}_i \bar{H}_{\bar{f}}^{-1}]^T, \; 0 \le i \le \bar{f} - 1, \tag{47}$$

and

$$\bar{E} = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & I \\ -\tilde{H}_0 & -\tilde{H}_1 & -\tilde{H}_2 & \cdots & -\tilde{H}_{\bar{f}-1} \end{bmatrix}. \tag{48}$$

We then compute the C^l-invariant subspace of the matrix $\bar{E}$, say Im $\bar{T}$, i.e.,

$$\bar{E}\bar{T} = \bar{T}\bar{E}_1,$$

where $\bar{E}_1 \in \mathbf{R}^{m \times m}$ and $\sigma(\bar{E}_1) \subset C^l$. Partitioning $\bar{T}$ into $m \times m$ blocks

$$\bar{T} = \begin{bmatrix} \bar{T}_1 \\ \bar{T}_2 \\ \vdots \\ \bar{T}_{\bar{f}} \end{bmatrix}, \tag{49}$$

we state an analogue of theorem 4 which gives the relationship between R_* and the matrix $\bar{T}$.

Theorem 7 *The matrix $\bar{T}_1$ is nonsingular and the matrix $\bar{X}$ defined by $\bar{X} = \bar{T}_1^{-T}\bar{T}_2^T$ equals R_0. Furthermore, the matrix $\bar{T}_1 - \bar{T}_2$ is nonsingular and the matrix $\bar{Y}$ defined by $\bar{Y} = (\bar{T}_1 - \bar{T}_2)^{-T}(\bar{T}_1 + \bar{T}_2)^T$ is the unique power-summable solution to $R = \bar{A}(R)$, i.e., $\bar{Y} = R_*$.*

7 ALGORITHMS

In the preceding sections, we have shown that solving for the M/G/1 or G/M/1 type Markov chains is equivalent to finding a certain invariant subspace of an associated matrix. Here, the invariance is with respect to the region C^{cl} in M/G/1 type Markov chains whereas it is with respect to the region C^{l} in G/M/1 type Markov chains. We now give the algorithms for solution of the nonlinear matrix equations (4) and (7) based on the matrix sign function described in Section 3.

In Section 5, solving for the unique power-bounded solution G_* to the matrix equation $G = A(G)$ is reduced to finding the C^{cl}-invariant subspace of the matrix E defined in (41). What we need is to be able to apply the results on the matrix sign function to compute this subspace. To provide this connection, we first note that the matrix E has a simple eigenvalue on the imaginary axis at $\lambda = 0$. Therefore, the matrix E cannot be used to start the matrix sign function iteration by definition of the matrix sign. For this purpose, we define x^T and y to be the left and right eigenvectors of E associated with the eigenvalue $\lambda = 0$, i.e.,

$$x^T E = 0, \; Ey = 0. \tag{50}$$

We also define

$$E_m = E - \frac{yx^T}{x^T y}. \tag{51}$$

By the above transformation, the simple eigenvalue of E at $\lambda = 0$ is now moved to $\lambda = -1$ preserving the eigenstructure. It is not difficult to show that the C^{cl}-invariant subspace of the matrix E and the left invariant subspace of E_m are equal. However, the matrix E_m does not have any eigenvalues on the imaginary axis and it can be used to start the matrix sign function iterations.

Moreover, the vectors x^T and y can easily be computed via observing:

$$\begin{aligned}
\hat{H}_0 &= H_f^{-1} H_0 \;\text{ by (40)} \\
&= H_f^{-1} F(1) \;\text{ by (35)} \\
&= H_f^{-1} D(1)[I - A(1)] \;\text{ by (31).}
\end{aligned}$$

The matrix $D(1)$ is nonsingular since $D(z)$ is analytic in the closed unit disk. Since $A(1)$ is stochastic and irreducible, $\hat{H}_0$ is singular and has a rank deficiency of one. Therefore, there exist nonzero vectors x_0^T and y_0 (unique up to a constant) such that

$$x_0^T \hat{H}_0 = 0, \ \hat{H}_0 y_0 = 0. \tag{52}$$

Note that since $A(1)$ is stochastic, there exists a probability vector $\hat{x}_0^T$ satisfying $\hat{x}_0^T A(1) = \hat{x}_0^T$. Furthermore, $A(1)e = e$ where we recall that e is a column vector of ones. Then the choice of $y_0 = e$ and $x_0^T = \hat{x}_0^T D(1)^{-1} H_f$ satisfy the equations (52). On a closer look at the definition (41) of the matrix E, the vectors defined by

$$x^T = \left[\begin{array}{ccccc} x_0^T \hat{H}_1 & x_0^T \hat{H}_2 & \cdots & x_0^T \hat{H}_{f-1} & x_0^T \end{array} \right] \tag{53}$$

and

$$y = \left[\begin{array}{c} y_0 \\ 0 \\ \vdots \\ 0 \end{array} \right] \tag{54}$$

satisfy (50).

Using theorem 4, and the properties of the matrix sign function, we now give the overall procedure to solve for G_* in Table 2, starting with a left coprime fraction for $A(z) = D^{-1}(z)N(z)$.

For G/M/1 type Markov chain, we analogously define

$$\bar{E}_m = \bar{E} + \frac{\bar{y}\bar{x}^T}{\bar{x}^T \bar{y}}, \tag{55}$$

where $\bar{E}$ is defined in (48) and $\bar{x}^T$ and $\bar{y}$ are the left and the right eigenvectors of $\bar{E}$ associated with the eigenvalue at $\lambda = 0$. It is not difficult to show that the left invariant subspaces of the two matrices $\bar{E}$ and $\bar{E}_m$ are equal. In order to find the left invariant subspace of $\bar{E}$ which gives R_*, we can therefore compute the matrix sign of $\bar{E}_m$ which is well-defined since the eigenvalue at $\lambda = 0$ of $\bar{E}$ is moved to the right half plane, i.e., to $\lambda = 1$. We also point out that the matrix $\tilde{H}_0$ defined in (47) has a rank deficiency of one and if $\bar{x}_0^T$ and $\bar{y}_0$ are defined so that

$$\bar{x}_0^T \tilde{H}_0 = 0, \ \tilde{H}_0 \bar{y}_0 = 0,$$

then the vectors

$$\bar{x}^T = \left[\begin{array}{ccccc} \bar{x}_0^T \tilde{H}_1 & \bar{x}_0^T \tilde{H}_2 & \cdots & \bar{x}_0^T \tilde{H}_{\bar{f}-1} & \bar{x}_0^T \end{array} \right] \tag{56}$$

Algorithm for finding G_*

---

1. Define $F(z) = zD(z) - N(z),\ F(z) = \sum_{i=0}^{f} F_i z^i$ as in (29).

2. Define $H(s) = \sum_{i=0}^{f} F_i(1-s)^{f-i}(1+s)^i = \sum_{i=0}^{f} H_i s^i$ as in (36).

3. Define $\hat{H}_i = H_f^{-1} H_i,\ 0 \le i \le f-1$ as in (40).

4. Find the nonzero vectors (unique up to multiplication by a constant) x_0^T, y_0 satisfying $x_0^T \hat{H}_0 = 0$ and $\hat{H}_0\, y_0 = 0$ and define x^T and y as in (53) and (54).

5. Define the $mf \times mf$ matrix E as in (41) and obtain the matrix $E_m = E - \frac{yx^T}{x^T y}$.

6. Find $Z = sgn(E_m)$ (see Section 3).

7. Obtain the $mf \times m$ matrix T by the m linearly independent columns of the matrix $(Z - I)$ that form a basis for the C^l-invariant subspace of E_m.

8. Partition T as in (42) and write $G_* = (T_1 + T_2)(T_1 - T_2)^{-1}$.

Table 2: The matrix sign based algorithm to find G_*

and

$$\bar{y} = \begin{bmatrix} \bar{y}_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{57}$$

are the left and the right eigenvectors of $\bar{E}$, respectively, associated with the eigenvalue at the origin.

We now give the overall procedure to obtain R_* starting from a right coprime fraction $A(z) = P(z)Q^{-1}(z)$ in Table 3. We note that, in the general Markov chains of M/G/1 and G/M/1 type, the computational cost of our algorithms exploiting matrix sign iterations is $O(m^3 f^3)$ for each iteration where m is the number of phases and f is defined as in (30).

Algorithm for finding R_*

---

1. Define $\bar{F}(z) = zQ(z) - P(z)$, $\bar{F}(z) = \sum_{i=0}^{\bar{f}} \bar{F}_i z^i$ as in (44).

2. Define $\bar{H}(s) = \sum_{i=0}^{\bar{f}} \bar{F}_i (1-s)^{\bar{f}-i}(1+s)^i = \sum_{i=0}^{\bar{f}} \bar{H}_i s^i$ as in (45).

3. Define $\tilde{H}_i = [\bar{H}_i \bar{H}_{\bar{f}}^{-1}]^T$, $0 \le i \le \bar{f} - 1$ as in (47).

4. Find the nonzero vectors (unique up to a constant) $\bar{x}_0^T, \bar{y}_0$ satisfying $\bar{x}_0^T \tilde{H}_0 = 0$ and $\tilde{H}_0 \bar{y}_0 = 0$ and define $\bar{x}^T$ and $\bar{y}$ as in (56) and (57).

5. Define the $m\bar{f} \times m\bar{f}$ matrix $\bar{E}$ as in (48) and obtain the matrix $\bar{E}_m = \bar{E} + \frac{\bar{y}\bar{x}^T}{\bar{x}^T \bar{y}}$.

6. Find $Z = sgn(\bar{E}_m)$ (see Section 3).

7. Obtain the $m\bar{f} \times m$ matrix $\bar{T}$ by the m linearly independent columns of the matrix $(Z - I)$ that form a basis for the C^l-invariant subspace of $\bar{E}_m$.

8. Partition $\bar{T}$ as in (49) and write $R_* = (\bar{T}_1 - \bar{T}_2)^{-T}(\bar{T}_1 + \bar{T}_2)^T$.

Table 3: The matrix sign based algorithm to find R_*

8 NUMERICAL EXAMPLES

In this section, we present a set of probability models by which we compare the algorithms proposed in this paper with the ones existing in the open literature in terms of execution times and precision. All algorithms are implemented in C using the CLAPACK library [46] on a DECAlphaStation 400 4/233 supporting DEC OSF/1 V3.2a. The CLAPACK library was built using a Fortran to C conversion utility called $f2c$ [16] for the Fortran 77 LAPACK (Linear Algebra PACKage) library [46] which includes subroutines for solving the most commonly occurring problems in numerical linear algebra. In implementing the linear algebra routines, the IEEE standard double precision arithmetic with machine precision $\epsilon \approx 2.2 \times 10^{-16}$ is used.

Example 1. We first examine a QBD chain composed of two phases considered in [35]. This example is found to be numerically ill-conditioned for some parameter values [35]. The QBD chain is described as follows: From the states $(i, 1)$ for $i \ge 1$, the chain moves to $(i, 2)$ with probability p and to $(i-1, 1)$ with probability $1-p$; from the states $(i, 2)$, $i \ge 0$, the chain moves to $(i,1)$ with probability $2p$ and to $(i+1,2)$ with probability $1-2p$. The QBD is ergodic for any value $p>0$. Considering the QBD chain as an M/G/1 type Markov chain as in (1), the probability generating matrix $A(z)$ of the Markov chain is written as:

$$A(z)=\underbrace{\begin{bmatrix} 1-p & 0 \\ 0 & 0 \end{bmatrix}}_{A_0}+\underbrace{\begin{bmatrix} 0 & p \\ 2p & 0 \end{bmatrix}}_{A_1}z+\underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1-2p \end{bmatrix}}_{A_2}z^2.$$

It can be checked that the G_* matrix for this QBD chain is:

$$G_*=\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}.$$

We employ the matrix sign function iteration (19) with scaling given in (21) in our procedure in Section 7 and the logarithmic reduction procedure of Latouche and Ramaswami [35] for finding the matrix G_* for performance comparison. We use the abbreviation *msf* for the former procedure whereas we use *log* for the latter algorithm. The procedure proposed in [35] has a quadratic convergence rate as the matrix sign function approach and to the best of our knowledge, it is one of the most time-efficient methods existing in the QBD literature. We use the stopping criterion (22) for algorithm *msf* and take $\varepsilon=10^{-8}$. We use the same value of ε in the stopping criterion $||Ge-e||_\infty<\varepsilon$ used in algorithm *log* in [35]. We have limited the maximum number of iterations in both algorithms to $maxit=40$. We use I, T, and err to denote the number of iterations required, the CPU time (in seconds) and the error in terms of $||G-G_*||_\infty$. The CPU times are obtained through averaging the outcomes of 2000 independent runs. Varying the traffic parameter p as in the way given in [35], we present our results on execution time and precision in Table 4. Examining Table 4, the algorithm *msf* appears to be slightly faster than algorithm *log*, the gain in CPU time when using *msf* increases with decreasing p. The procedure *msf* captures the G_* matrix with an accuracy close to machine precision for all values of p whereas the accuracy of logarithmic reduction technique appears to deteriorate as the parameter p decreases. The next example is also a QBD chain with 24 phases by which it is more appropriate to draw conclusions about performances of both algorithms in terms of execution time and accuracy.

Example 2. The example we now present has been examined in both [14] and [35]. The problem considered in these references is a teletraffic

p	I_{msf}	I_{log}	T_{msf}	T_{log}	err_{msf}	err_{log}
10^{-1}	6	7	$1.7\ 10^{-3}$	$1.4\ 10^{-3}$	$9.2\ 10^{-16}$	$5.0\ 10^{-14}$
10^{-2}	7	10	$1.7\ 10^{-3}$	$1.8\ 10^{-3}$	$3.6\ 10^{-16}$	$4.8\ 10^{-10}$
10^{-3}	8	14	$1.9\ 10^{-3}$	$2.4\ 10^{-3}$	$1.1\ 10^{-15}$	$1.1\ 10^{-13}$
10^{-4}	8	17	$1.8\ 10^{-3}$	$2.9\ 10^{-3}$	$8.0\ 10^{-16}$	$1.9\ 10^{-12}$
10^{-5}	8	20	$1.7\ 10^{-3}$	$3.5\ 10^{-3}$	$7.5\ 10^{-16}$	$3.7\ 10^{-10}$
10^{-6}	9	24	$2.1\ 10^{-3}$	$4.2\ 10^{-3}$	$6.4\ 10^{-16}$	$1.2\ 10^{-10}$
10^{-7}	9	27	$2.1\ 10^{-3}$	$4.7\ 10^{-3}$	$1.3\ 10^{-15}$	$9.6\ 10^{-11}$
10^{-8}	9	>40	$2.1\ 10^{-3}$	$6.5\ 10^{-3}$	$1.1\ 10^{-15}$	$1.5\ 10^{-8}$

Table 4: Number of iterations, CPU time and error for algorithms msf and log for Example 1 with $\varepsilon = 10^{-8}$.

system which is a continuous-time QBD with matrices of order $m = 24$. The infinitesimal generator matrix for this QBD is

$$Q = \begin{bmatrix} B' & A'_0 & 0 & 0 & \cdots \\ A'_2 & A'_1 & A'_0 & 0 & \cdots \\ 0 & A'_2 & A'_1 & A'_0 & \cdots \\ 0 & 0 & A'_2 & A'_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where A'_0 and A'_2 are diagonal with $A'_0 = 192\,\rho_d\,I$ and $(A'_2)_{jj} = 192\,(1 - j/24)$, $0 \le j \le 23$. The matrix A'_1 is tridiagonal, with $(A'_1)_{j,j+1} = a\,r\,(M - j)/M$, $0 \le j \le 22$, and $(A'_1)_{j,j-1} = j\,r$, $1 \le j \le 23$. The matrix B' is equal to $A'_1 + A'_0$. The parameters ρ_d, a, r, and M are quantities related with the teletraffic model. It is also given in [35] how to consider the embedded process at epochs of change of level from which one can obtain the G_* matrix for the embedded Markov chain (QBD chain) which has indeed the same G_* of the original continuous-time QBD.

Besides obtaining the G_* matrix, we also find the same queue statistics as in [14], namely, probability of queue length being empty and mean queue length using the algorithm msf as well as the error (err_{msf} and err_{log}) in terms of $||Ge - e||_\infty$. We present our results on execution time and queue statistics in Table 5 for the first set of problem parameter values given in [14]. We obtain the same seven significant digits as reported in [14] except for the cases (marked by $*$) where there is a difference in some of last digits. Except for the ill-conditioned cases of $M = 32768$ and 65536 for which the algorithm log does not satisfy the stopping criterion within $maxit$ iterations, two algorithms exhibit similar performances in terms of execution time. The precision of algorithm msf appears to be better than

M	I_{msf}	I_{log}	T_{msf}	T_{log}	err_{msf}	err_{log}	$P[empty]$	$E[queue]$
64	9	16	$3.7\ 10^{-1}$	$7.3\ 10^{-1}$	$1.1\ 10^{-15}$	$2.2\ 10^{-12}$	$2.575932\ 10^{-1}$	$1.172273\ 10^{3}$
128	9	17	$3.5\ 10^{-1}$	$8.0\ 10^{-1}$	$1.1\ 10^{-15}$	$1.5\ 10^{-10}$	$1.325646\ 10^{-1}$	$6.568136\ 10^{3}$
256	10	18	$4.0\ 10^{-1}$	$8.3\ 10^{-1}$	$3.8\ 10^{-15}$	$5.3\ 10^{-10}$	$6.551767\ 10^{-2}$	$1.974862\ 10^{4}$
512	11	19	$4.2\ 10^{-1}$	$9.0\ 10^{-1}$	$1.1\ 10^{-14}$	$8.0\ 10^{-10}$	$3.227063\ 10^{-2}$	$4.738760\ 10^{4}$
1024	12	20	$4.3\ 10^{-1}$	$9.3\ 10^{-1}$	$3.4\ 10^{-14}$	$6.2\ 10^{-10}$	$1.589722\ 10^{-2}$	$1.038203\ 10^{5}$
2048	13	21	$5.2\ 10^{-1}$	$9.8\ 10^{-1}$	$2.5\ 10^{-13}$	$8.5\ 10^{-11}$	$7.793980\ 10^{-3}$	$2.196087\ 10^{5}$
4096	14	22	$5.2\ 10^{-1}$	$1.0\ 10^{0}$	$3.4\ 10^{-13}$	$1.8\ 10^{-10}$	$3.765687\ 10^{-3}$	$4.626801\ 10^{5}$
8192	15	23	$5.2\ 10^{-1}$	$1.1\ 10^{0}$	$2.6\ 10^{-12}$	$1.8\ 10^{-10}$	$1.757668\ 10^{-3}$	$1.000001\ 10^{6}$
16384	16	24	$5.7\ 10^{-1}$	$1.1\ 10^{0}$	$1.6\ 10^{-11}$	$9.8\ 10^{-9}$	$7.552270\ 10^{-4}$	$2.337514\ 10^{6}$
32768	17	> 40	$6.0\ 10^{-1}$	$1.7\ 10^{0}$	$9.1\ 10^{-11}$	$1.3\ 10^{-8}$	$2.544028\ 10^{-4}$	$*6.954319\ 10^{6}$
65536	27	> 40	$8.8\ 10^{-1}$	$1.8\ 10^{0}$	$6.2\ 10^{-7}$	$3.2\ 10^{-7}$	$*4.090361\ 10^{-6}$	$*4.337820\ 10^{8}$

Table 5: Comparison of algorithms msf and log for Example 2 with $\varepsilon = 10^{-8}$, $r = 300^{-1}$, $a = 18.244$, $\rho_d = 0.280$.

that of log for all values of M we have tried although unlike the previous example, we observe deterioration in precision for the algorithm msf with increasing M.

We then use another example taken from [14] and [35] for the same model (see Table 6) with a different set of traffic parameters. In this case, when $\rho_d > 0.05$, the algorithm msf appears to outperform the algorithm log in terms of both execution time and precision. We note that in the last entry of the table, we take $\rho_d = 0.29568$ to obtain the same emptiness probability as in [14] up to six significant digits instead of $\rho_d = 0.296$ (displayed up to three significant digits in both two references) which otherwise yields an unstable queueing system. We also note that the discrepancy between the numerical results obtained in [35] and our implementation of the logarithmic reduction procedure for Examples 1 and 2 stem from the difference in the computing platforms employed.

Next, we present a numerical example for an M/G/1 type Markov chain with a general rational probability generating matrix.

Example 3. In this example, we examine a number of algorithms when the pgm $A(z)$ is rational for which the solution to $G = A(G)$ is of concern. The matrix sign iteration based method msf and the following four algorithms will be considered.

(1) $G_0 = 0,\ G_{j+1} = (I - A_1)^{-1}(A_0 + \sum_{i=2}^{\infty} A_i G_j^i)$, based on [42].

(2) $G_0 = 0,\ G_{j+1} = (I - \sum_{i=1}^{\infty} A_i G_j^{i-1})^{-1} A_0$, given in [44].

ρ_d	I_{msf}	I_{log}	T_{msf}	T_{log}	err_{msf}	err_{log}	$P[empty]$	$E[queue]$
.01000	9	3	3.7 10^{-1}	1.5 10^{-1}	4.4 10^{-16}	2.4 10^{-10}	9.495388 10^{-1}	5.607618 10^{-2}
.02500	9	4	3.8 10^{-1}	2.0 10^{-1}	6.7 10^{-16}	9.7 10^{-9}	8.741457 10^{-1}	1.774883 10^{-1}
.05000	9	7	3.7 10^{-1}	3.5 10^{-1}	6.7 10^{-16}	5.8 10^{-10}	7.533826 10^{-1}	1.010752 10^{0}
.07500	9	10	3.7 10^{-1}	4.7 10^{-1}	6.7 10^{-16}	7.1 10^{-14}	6.448787 10^{-1}	6.150972 10^{0}
.10000	9	11	3.7 10^{-1}	5.3 10^{-1}	8.9 10^{-16}	4.1 10^{-12}	5.462015 10^{-1}	2.418806 10^{1}
.12000	9	12	3.7 10^{-1}	5.7 10^{-1}	1.1 10^{-15}	3.8 10^{-13}	4.731717 10^{-1}	5.779082 10^{1}
.14000	9	13	3.5 10^{-1}	6.0 10^{-1}	1.1 10^{-15}	3.4 10^{-13}	4.048339 10^{-1}	1.218496 10^{2}
.16000	9	13	3.7 10^{-1}	6.2 10^{-1}	1.3 10^{-15}	3.1 10^{-10}	3.408049 10^{-1}	2.366025 10^{2}
.18000	9	14	3.8 10^{-1}	6.7 10^{-1}	8.9 10^{-16}	1.1 10^{-12}	2.807915 10^{-1}	4.362660 10^{2}
.20000	9	14	3.7 10^{-1}	6.5 10^{-1}	1.1 10^{-15}	3.5 10^{-9}	2.245312 10^{-1}	7.831238 10^{2}
.22000	9	15	3.5 10^{-1}	7.0 10^{-1}	1.3 10^{-15}	4.9 10^{-12}	1.717838 10^{-1}	1.404345 10^{3}
.24000	9	16	3.5 10^{-1}	7.5 10^{-1}	1.3 10^{-15}	6.6 10^{-12}	1.223174 10^{-1}	2.697990 10^{3}
.26000	10	16	3.8 10^{-1}	7.5 10^{-1}	2.6 10^{-15}	4.6 10^{-9}	7.590597 10^{-2}	5.389706 10^{3}
.28000	11	18	4.2 10^{-1}	8.3 10^{-1}	2.7 10^{-15}	2.1 10^{-11}	3.233099 10^{-2}	1.581604 10^{4}
.29000	12	19	4.5 10^{-1}	8.8 10^{-1}	2.3 10^{-14}	1.2 10^{-10}	1.154133 10^{-2}	4.911922 10^{4}
.29500	15	22	5.3 10^{-1}	1.1 10^{0}	2.2 10^{-12}	4.8 10^{-10}	1.384957 10^{-3}	4.301674 10^{5}
.29568	20	>40	6.8 10^{-1}	1.7 10^{0}	1.5 10^{-8}	1.7 10^{-8}	1.572177 10^{-5}	$*$3.814722 10^{7}

Table 6: Number of iterations, CPU time and error of algorithms msf and log for Example 2 with $\varepsilon = 10^{-8}$, $r = 100^{-1}$, $a = 18.244$, $M = 512$.

(3) An extension of iteration (1) which takes advantage of the reduced polynomial equation $F(G) = 0$ in (32) and therefore does not need truncation (assuming invertibility of F_1):

$$G_0 = 0,\ G_{j+1} = -F_1^{-1}(F_0 + \sum_{i=2}^{f} F_i G_j^i),$$

(4) An extension of iteration (2):

$$G_0 = 0,\ G_{j+1} = (-\sum_{i=1}^{f} F_i G_j^{i-1})^{-1} F_0.$$

Remarks:

- We note that the algorithms (1)-(4) we use for comparison all have linear convergence rates and we do not perform any comparisons in the paper with the existing quadratically convergent algorithms for M/G/1 type chains proposed in [34],[36], and [8].

- We have shown in Section 5 that when the pgm $A(z)$ is a rational matrix, the matrix equation $G - A(G) = 0$ can be reduced to a matrix polynomial equation $F(G) = 0$. One can then naturally extend the fixed-point iterations (1) and (2) so that the infinite summation at each iteration is replaced with an appropriate matrix polynomial evaluation. That is how we obtain the modified iterations (3) and (4).

- In iterations (1) and (2), a truncation index K is used such that $||A_i||_\infty < \varepsilon_0$, $\forall i > K$, see [42]. In implementation of these algorithms, one may replace the upper limit of summation in iterations (1) and (2) by the truncation index K.

- In all the iterations above, we use the same stopping criterion $||Ge - e||_\infty < \varepsilon$, where the same value of ε is also used in the stopping criterion as for the matrix sign function iterations.

- We note that the computational cost at each step of iterations (1)-(4) is $O(m^3 f)$ compared with $O(m^3 f^3)$ computational cost of the matrix sign based algorithms, however the latter algorithms have *at least* quadratic convergence rates as compared to the linear convergence rates of the algorithms (1)-(4).

In order to make the comparisons, we take a Markov chain of M/G/1 type with the following rational probability generating matrix:

$$A(z) = \begin{bmatrix} 0.0002 & 0.9998 \\ 0.9800 & 0.0200 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \frac{1-a}{1-az} \end{bmatrix}.$$

and vary the parameter, a, to obtain different utilizations (ρ) where ρ is defined by $xA'(1)e$, x being the invariant vector of $A(1)$ [42]. We use $\varepsilon = 10^{-8}$ and $\varepsilon_0 = 10^{-10}$ in our comparisons. We present our results on the CPU time and number of iterations required to solve G_* in Tables 7 and 8 with respect to the utilization of the system.

We summarize our observations:

- The algorithm (2) given in [44] performs better than the other traditional iteration (1) in terms of the number of iterations and CPU time particularly when utilization is not high.

- The algorithms (3) and (4) that take advantage of the rational structure of $A(z)$ and the reduced equation (32) provide a uniform gain in CPU time over the algorithms (1) and (2). This is because the

ρ	(1)	(2)	(3)	(4)	msf
.20	8	6	18	16	6
.40	14	10	35	28	6
.60	26	18	67	50	6
.80	57	38	158	112	6
.90	115	78	333	228	7
.95	227	152	665	449	7
.99	1127	753	3080	2055	7

Table 7: Number of iterations of algorithms (1)-(4) and msf for Example 3.

ρ	(1)	(2)	(3)	(4)	msf
.20	$6.7\ 10^{-3}$	$2.4\ 10^{-3}$	$6.7\ 10^{-4}$	$2.2\ 10^{-3}$	$2.6\ 10^{-3}$
.40	$6.7\ 10^{-3}$	$4.8\ 10^{-3}$	$1.7\ 10^{-3}$	$2.2\ 10^{-3}$	$2.6\ 10^{-3}$
.60	$2.0\ 10^{-2}$	$1.7\ 10^{-2}$	$2.7\ 10^{-3}$	$5.6\ 10^{-3}$	$2.6\ 10^{-3}$
.80	$5.3\ 10^{-2}$	$4.8\ 10^{-2}$	$6.3\ 10^{-3}$	$1.6\ 10^{-2}$	$2.5\ 10^{-3}$
.90	$1.2\ 10^{-1}$	$1.2\ 10^{-1}$	$1.3\ 10^{-2}$	$3.3\ 10^{-2}$	$2.7\ 10^{-3}$
.95	$2.4\ 10^{-1}$	$3.6\ 10^{-1}$	$2.5\ 10^{-2}$	$7.6\ 10^{-2}$	$2.8\ 10^{-3}$
.99	$1.2\ 10^{0}$	$4.7\ 10^{0}$	$1.2\ 10^{-1}$	$5.8\ 10^{-1}$	$2.8\ 10^{-3}$

Table 8: CPU times of algorithms (1)-(4) and msf for Example 3.

computation at each iteration is reduced significantly compared to (1) and (2) although more iterations are required for (3) and (4).

- The algorithm msf generally outperforms all other iterations for all but low utilizations. The gain in using msf increases with increasing load for which other iterations suffer from linear convergence rates.

- The accuracy of algorithm msf was better ($||Ge - e||_\infty < 10^{-12}$ for all cases) than the others (error was in the order of 10^{-9} for all cases).

Based on these observations, we conclude that the matrix sign function based algorithm msf gives the best performance for both CPU time and numerical precision for M/G/1 type Markov chains where the degree of the polynomial matrix $F(z)$ turns out to be $f = 2$. We also have conducted other numerical experimentations to compare the algorithms (3) and (4) with the algorithm msf when f is a variable parameter whose results we do not include in the current paper due to space limitations (see [1] and [2] for more details). In these examples, we mainly observe that if we fix the utilization, then there is a cross-over value of f below which the algorithm msf outperforms the others in terms of execution time.

On the other hand, when the utilization increases, this cross-over value significantly increases favoring the matrix sign function approach.

9 CONCLUSIONS

In this paper, novel algorithms are proposed for the solution of nonlinear matrix equations arising in M/G/1 and G/M/1 type Markov chains. We show that the solution of these equations can be reduced to computing a basis for the left invariant subspace of a real matrix when the underlying probability generating matrix $A(z)$ is rational. This covers a rich set of stochastic models. Such invariant subspace computations can effectively be performed using a wide variety of techniques including the matrix sign function iterations which can be implemented with quadratic or higher convergence rates.

We concentrated on the particular iteration (19) with scaling given in (21) in our numerical examples. We have used the standard LAPACK [46] library in implementing the algorithms. We have shown through numerical examples that in the case of QBD chains (a subcase for the algorithms we propose), the matrix sign function based algorithm slightly outperforms the logarithmic reduction procedure [35] both in terms of execution time and precision, the latter designed for only QBD chains and being one of the most time-efficient algorithm we are aware of. We note that the particular matrix sign iteration we employed in our numerical experimentation and the algorithm reported in [35] both have quadratic convergence rates.

In the general Markov chains of M/G/1 type, the computational cost of the algorithms exploiting matrix sign iterations is $O(m^3 f^3)$ for each step where m is the number of phases and degree f is defined as in (30) compared to the $O(m^3 f)$ cost for each iteration of linear convergence rate algorithms (3) and (4). On the other hand, the traditional iterations (1) or (2) based on truncation have $O(m^3 K)$ computational cost at each step where K denotes the truncation index. The performance of the algorithms (3) and (4) in terms of CPU time appears to be better than (1) and (2), respectively, despite the requirement for more iterations to converg since the computation per iteration is significantly reduced using (3) and (4). Moreover, the high convergence rates of matrix sign iterations offer significant gains in computation time compared to all the other algorithms

(1)-(4) particularly when f is not large. The performance gain increases significantly with increasing load. We therefore conclude that the the matrix sign function based algorithms are very suitable for solving large scale M/G/1 and G/M/1 type Markov chains with high utilization and/or with low degree parameter f.

References

[1] N. Akar and K. Sohraby. A new paradigm in teletraffic analysis of communication networks. In *INFOCOM*, pages 1318–1326, 1996.

[2] N. Akar and K. Sohraby. On computational aspects of the invariant subspace approach to teletraffic problems and comparisons. Computer Science Telecommunications tech. report available in postscript form as ftp.cstp.umkc.edu:/public/papers/akar/comp.ps, Univ. Missouri - Kansas City, 1996.

[3] 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.

[4] Z. Bai and J. Demmel. 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.

[5] B. Bakkaloğlu, C. K. Koç, and M. İnceoğlu. Parallel matrix sign iterations using PVM. *IEEE Trans. Auto. Contr.*, 1995. submitted for publication.

[6] L. Balzer. Accelerated convergence of the matrix sign function. *Int. Jour. Contr.*, 32:1057–1078, 1980.

[7] D. Bini and B. Meini. On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems. In *Proceedings of the Second International Workshop on Numerical Solution of Markov Chains*, pages 21–38, Raleigh, North Carolina, 1995.

[8] D. Bini and B. Meini. On the solution of a nonlinear matrix equation in queueing problems. *SIAM J. of Matrix Anal. Appl.*, 1995. submitted for publication.

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

[10] E. Çinlar. Time dependence of queues with Semi-Markovian services. *J. Appl. Prob.*, 4:356–364, 1967.

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

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

[13] J. N. Daigle, Y. Lee, and M. N. Magalhaes. Discrete time queues with phase dependent arrivals. In *IEEE INFOCOM*, pages 728–732, 1992.

[14] J. N. Daigle and D. M. Lucantoni. Queueing systems having phase-dependent arrival and service rates. In *Numerical Solution of Markov chains*, pages 223–238, New York, 1991. Marcel Dekker Inc.

[15] E. D. Denman and A. N. Beavers. The matrix sign function and computation in systems. *Appl. Math. Comput.*, 2:63–94, 1976.

[16] S. I. Feldman, D. M. Gay, M. W. Maimone, and N. L. Schryer. A Fortran-to-C converter. Technical Report Computing Science Tech. Report No. 149, AT & T Bell Laboratories, 1990.

[17] H. R. Gail, S. L. Hantler, M. Sidi, and B. A. Taylor. Linear independence of root equations for M/G/1 type Markov chains. *Queueing Systems*, 20:321–339, 1995.

[18] H. R. Gail, S. L. Hantler, and B. A. Taylor. Solutions of the basic matrix equation for M/G/1 and G/M/1 type Markov chains. *Commun. Statist.-Stochastic Models*, 10(1):1–43, 1994.

[19] H. R. Gail, S. L. Hantler, and B. A. Taylor. Spectral analysis of M/G/1 and G/M/1 type Markov chains. to appear in Adv. Appl. Prob., 1996.

[20] 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.

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

[22] I. C. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials.* Academic Press, New York, 1982.

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

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

[25] L. Gün. Experimental techniques on matrix-analytical solution techniques - extensions and comparisons. *Commun. Statist. - Stochastic Models*, 5(4):669–682, 1989.

[26] N. J. Higham. The matrix sign decomposition and its relation to polar decomposition. *Lin. Alg. Appl.*, 212:3–20, 1994.

[27] N. J. Higham and P. Papadimitriou. A parallel algorithm for computing the polar decomposition. *Parallel Computing*, 20(8):1161–1173, 1994.

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

[29] K. Kant. *Introduction to Computer System Performance Evaluation.* McGraw-Hill, New York, 1992.

[30] C. Kenney and A. J. Laub. Rational iterative methods for the matrix sign function. *SIAM Matrix. Anal. Appl.*, 12(2):273–291, 1991.

[31] C. Kenney and A. J. Laub. On scaling Newton's method for polar decomposition and the matrix sign function. *SIAM Jour. Matrix Anal. Appl.*, 13(3):688–706, 1992.

[32] P. Lancaster and M. Tismenetsky. *The Theory of Matrices.* Academic Press, New York, 1985.

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

[34] G. Latouche. Newton's iteration for non-linear equations arising in Markov chains. *IMA Jour. of Num. Anal.*, 14:583–598, 1994.

[35] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for quasi-birth-death processes. *J. Appl. Prob.*, 30:650–674, 1993.

[36] G. Latouche and G. W. Stewart. Numerical methods for M/G/1 type queues. In *Proc. 2nd Int. Workshop on Numerical Solution of Markov Chains*, pages 571–581, Raleigh, NC, USA, 1995.

[37] A. J. Laub. Invariant subspace methods for the numerical solution of Riccati equations. In S. Bittanti, A. J. Laub, and J. C. Willems, editors, *The Riccati Equation*, chapter 7, pages 163–196. Springer-Verlag, Berlin, 1991.

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

[39] 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.

[40] M. F. Neuts. The single server queue with Poisson input and Semi-Markov service times. *J. Appl. Prob.*, 3:202–230, 1966.

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

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

[43] P. Pandey, C. Kenney, and A. J. Laub. A parallel algorithm for the matrix sign function. *International Jour. of High Speed Computing*, 2(2):181–191, 1990.

[44] V. Ramaswami. Nonlinear matrix equations in applied probability - solution techniques and open problems. *SIAM Review*, 30(2):256–263, 1988.

[45] 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.

[46] SIAM. *LAPACK Users' Guide*, 1995. also available in html form over the Internet as http://www.netlib.org/lapack.

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

[48] 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.

[49] J. Ye and S. Q. Li. Analysis of multi-media traffic queues with finite buffer and overload control- Part I: Algorithm. In *Proc. IEEE INFOCOM*, pages 1464–1474, 1991.

[50] J. Ye and S. Q. Li. Analysis of multi-media traffic queues with finite buffer and overload control- Part II: Applications. In *Proc. IEEE INFOCOM*, pages 848–859, 1992.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive comments which substantially improved the presentation of this manuscript. The authors also wish to thank A. Agrawal (UMKC) for writing the C code for implementing the algorithms.

This research was supported in part by University of Missouri Research Board (UMRB) grant K-3-40531 and by NSF under grants NCR-9508014 and CDA-9422092.